

160

Application	108
Transactional	106
Middleware	104
Network OS	102
Hardware	100

Figure 1A

180

Java	C +	Cobal	Small Talk	120
CORBA				124
Networks OS				102
Hardware				100

Figure 1C

170

Java	118
Enterprise Java Beans	114
Network OS	102
Hardware	100

Figure 1B

190

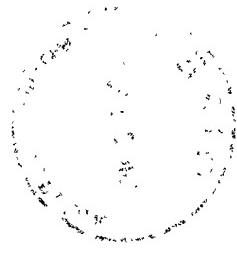
Java	C +	Cobal	Small Talk	120
Windows DNS (COM/MTS)				134
Network OS				102
Hardware				100

Figure 1D

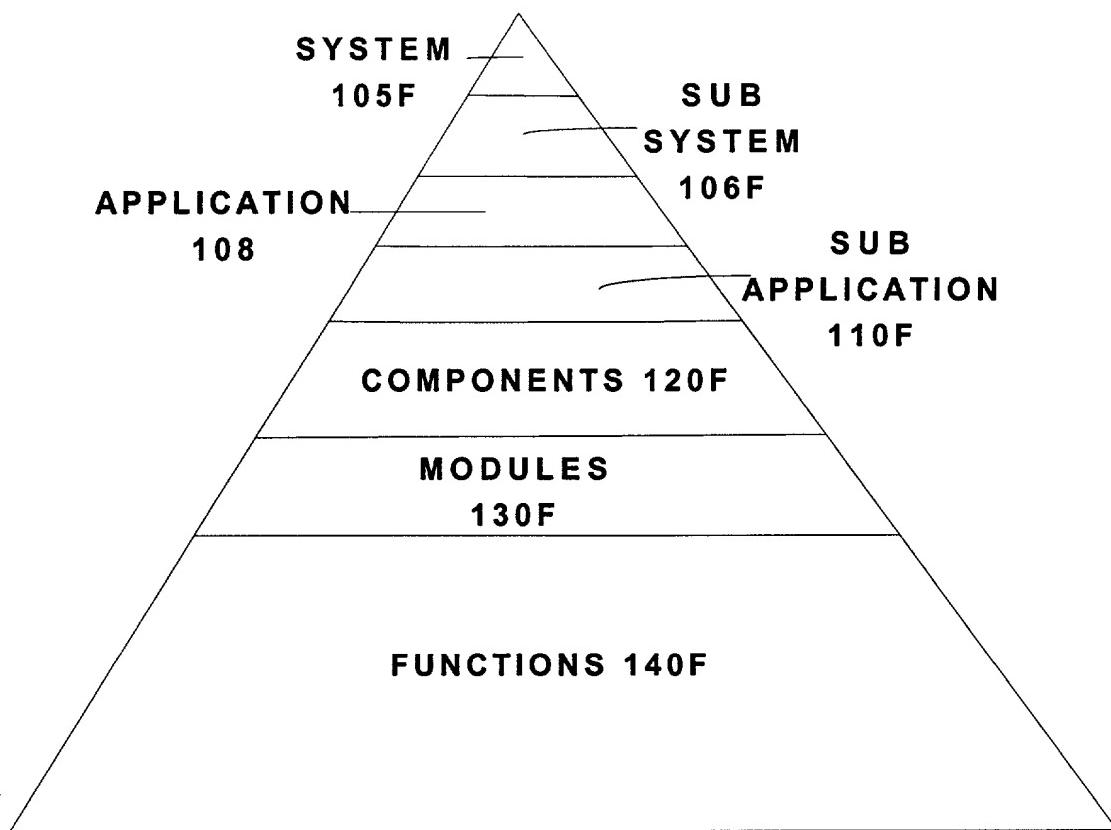
195

C	146
Tuxedo	144
Network OS	102
Hardware	100

Figure 1E



100F



PRIOR ART

FIGURE 1F

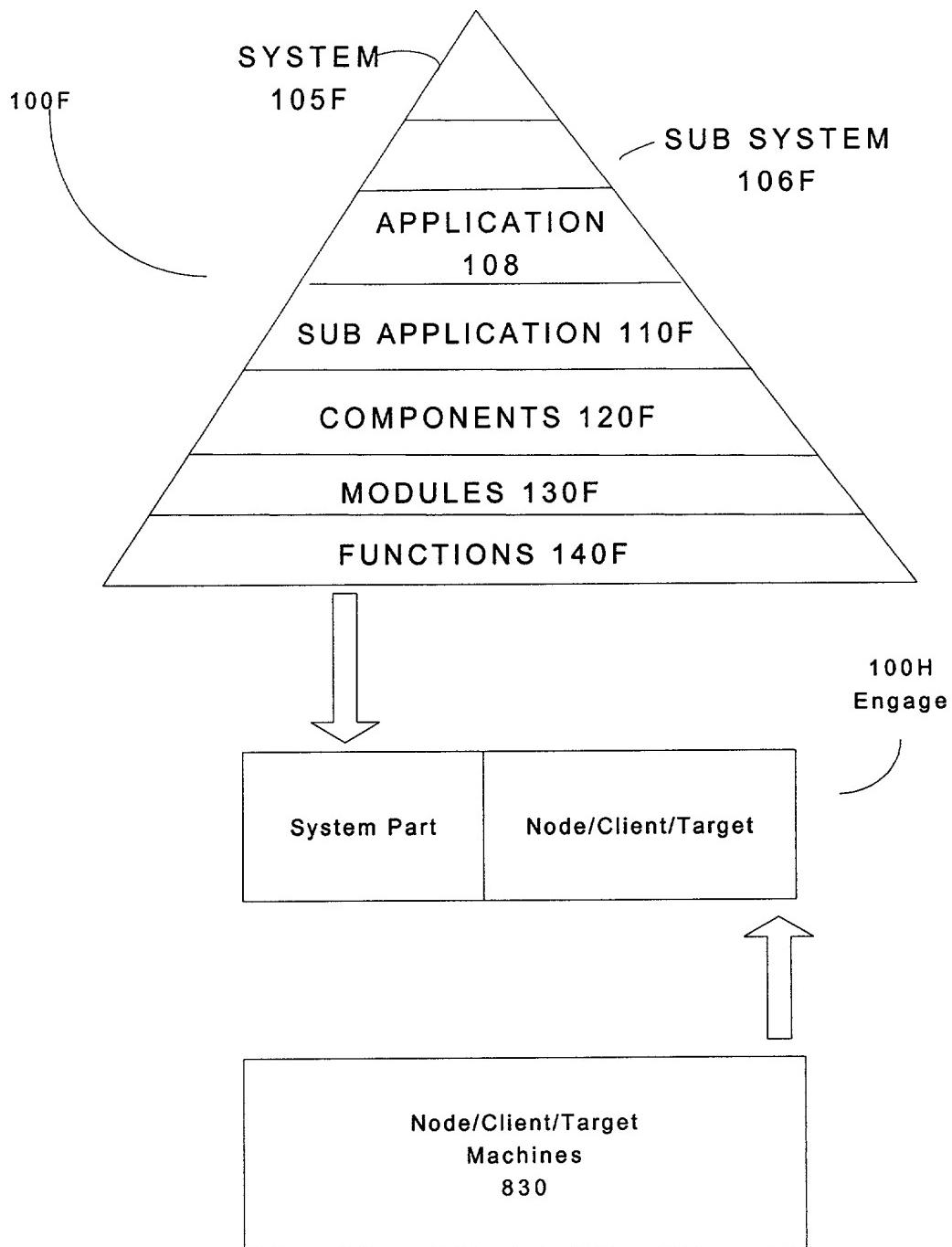


Figure 1G

100H

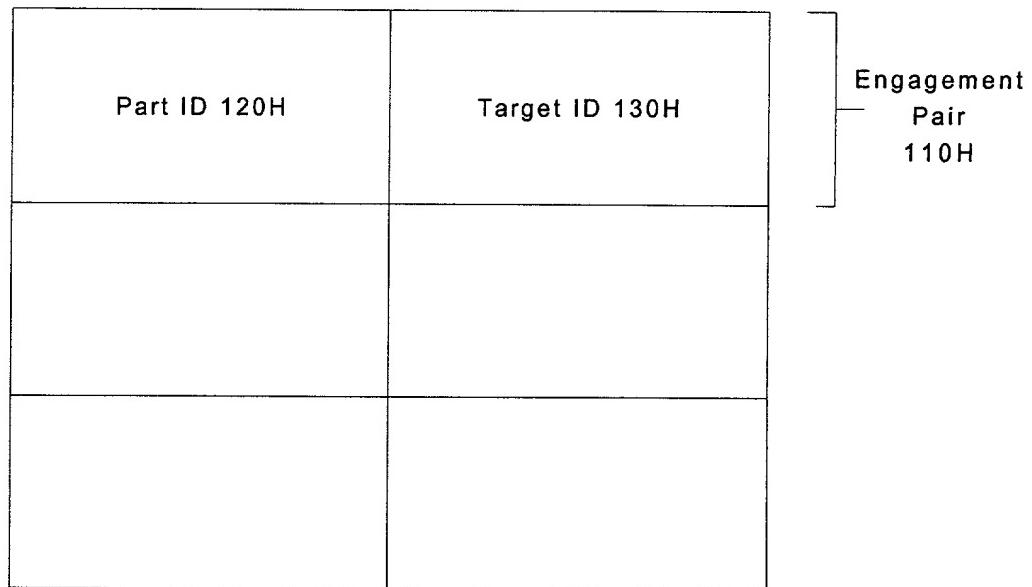


Figure 1H

200

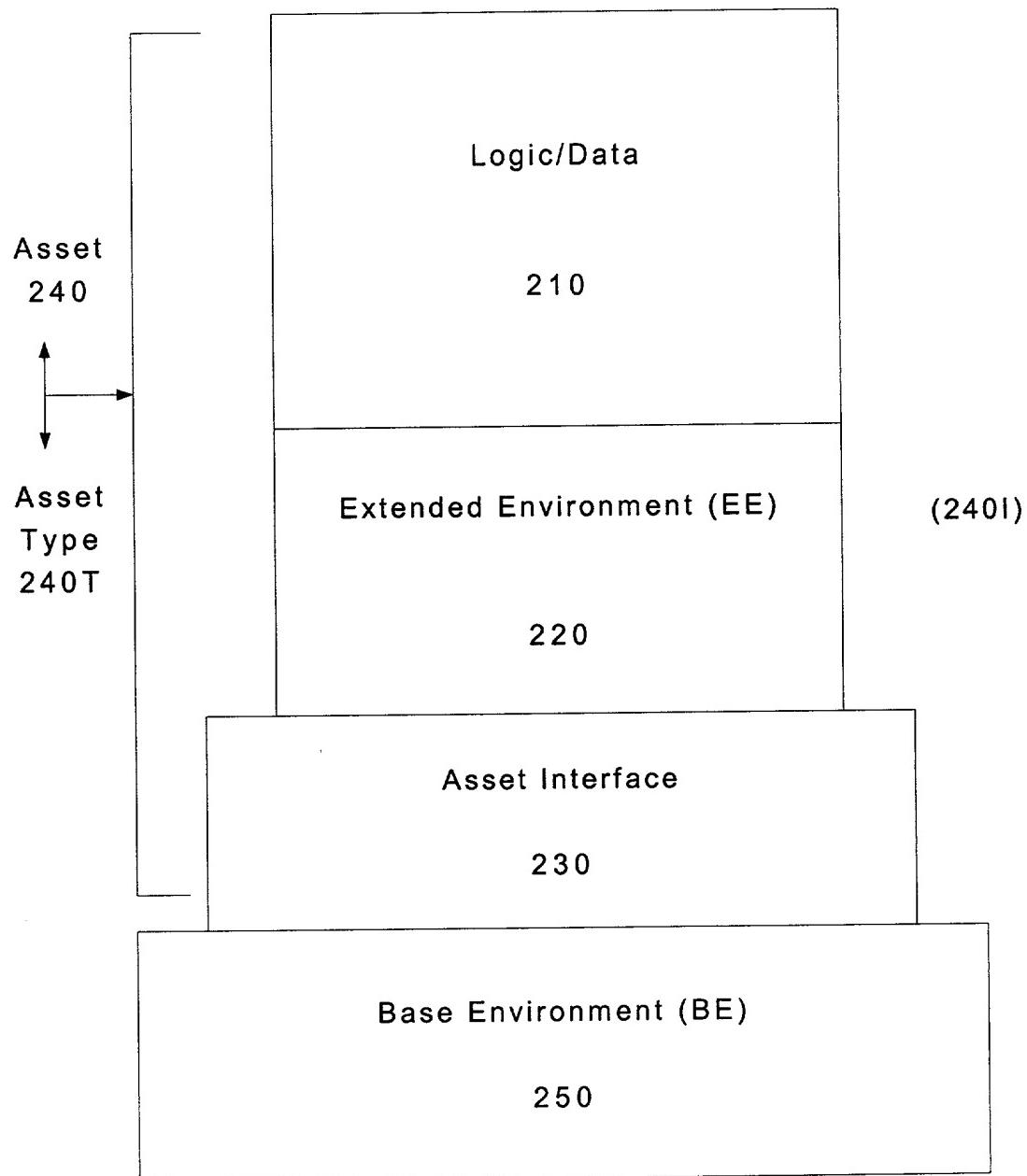


Figure 2

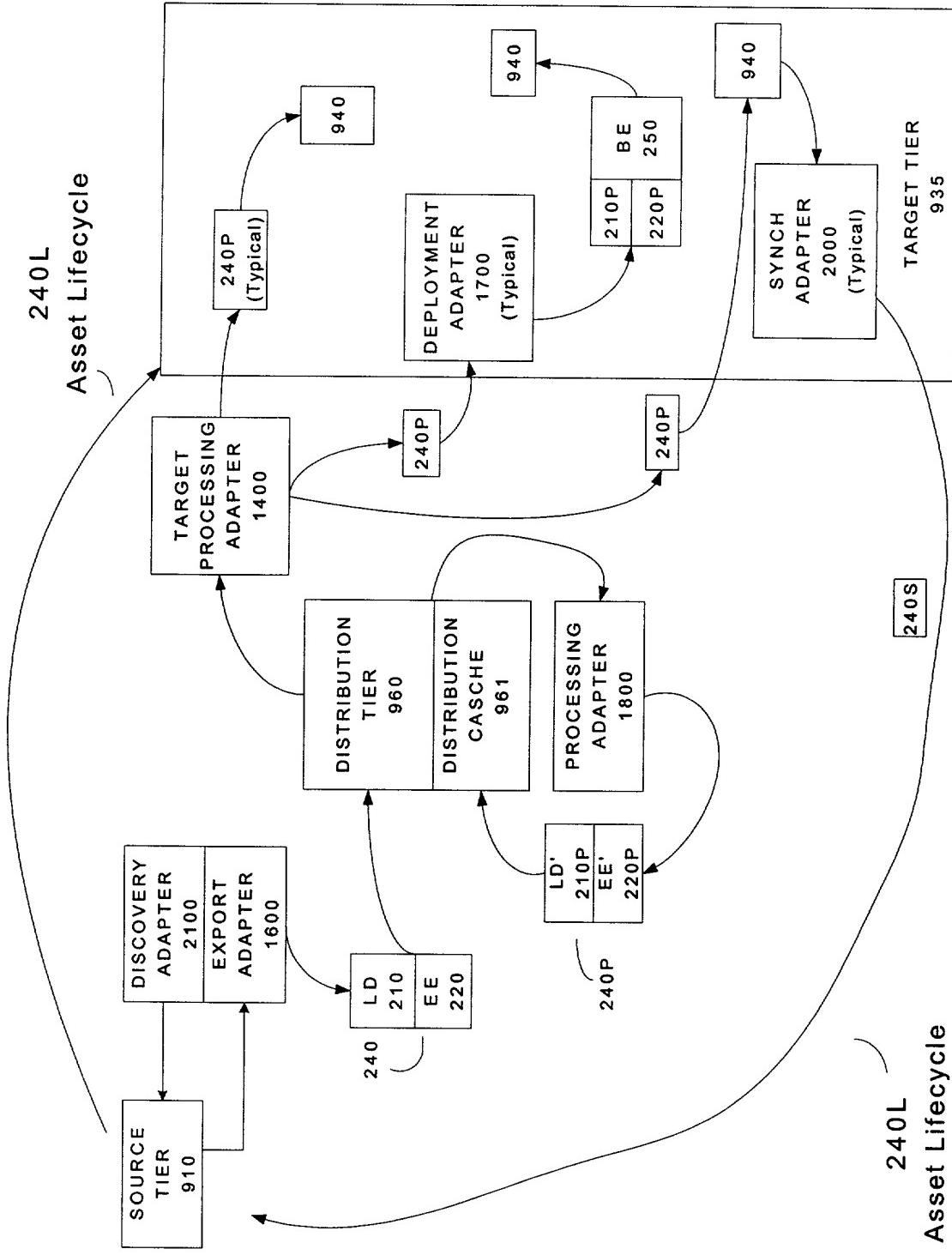


FIGURE 2A

220

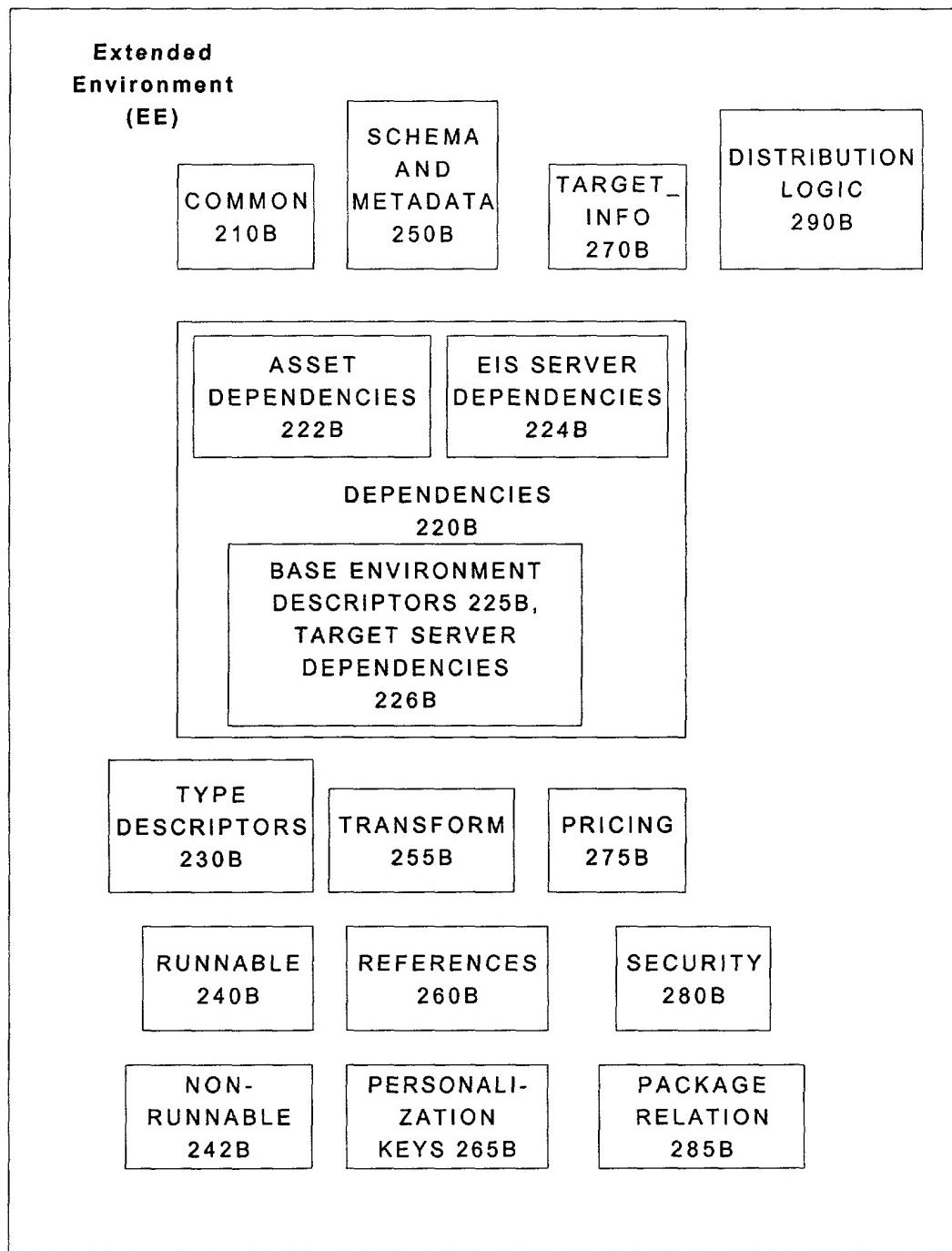


FIGURE 2B

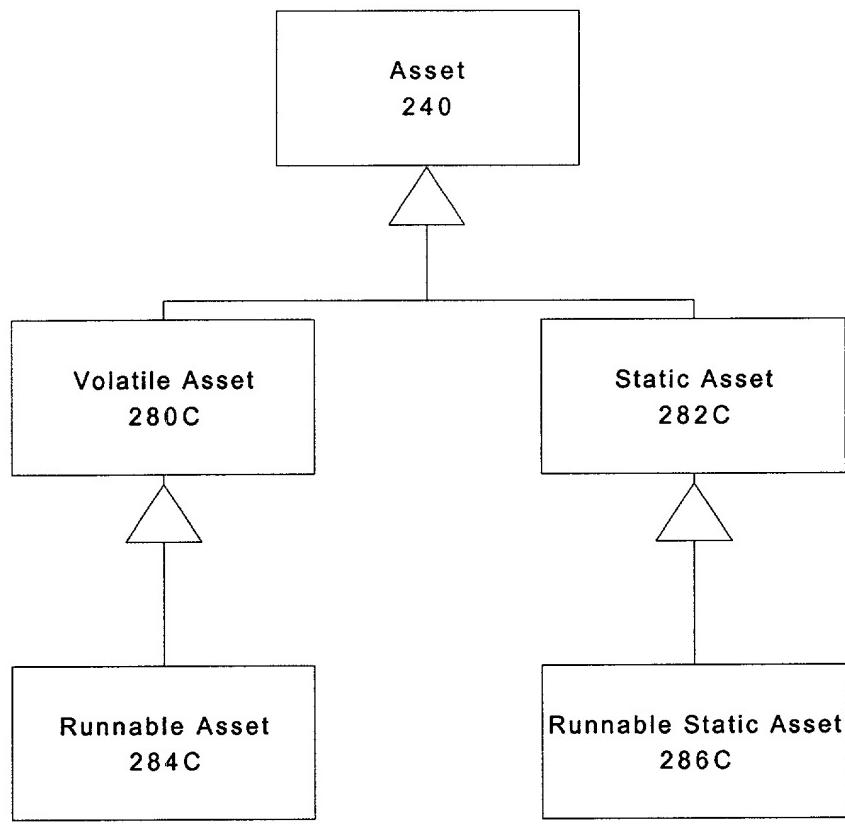
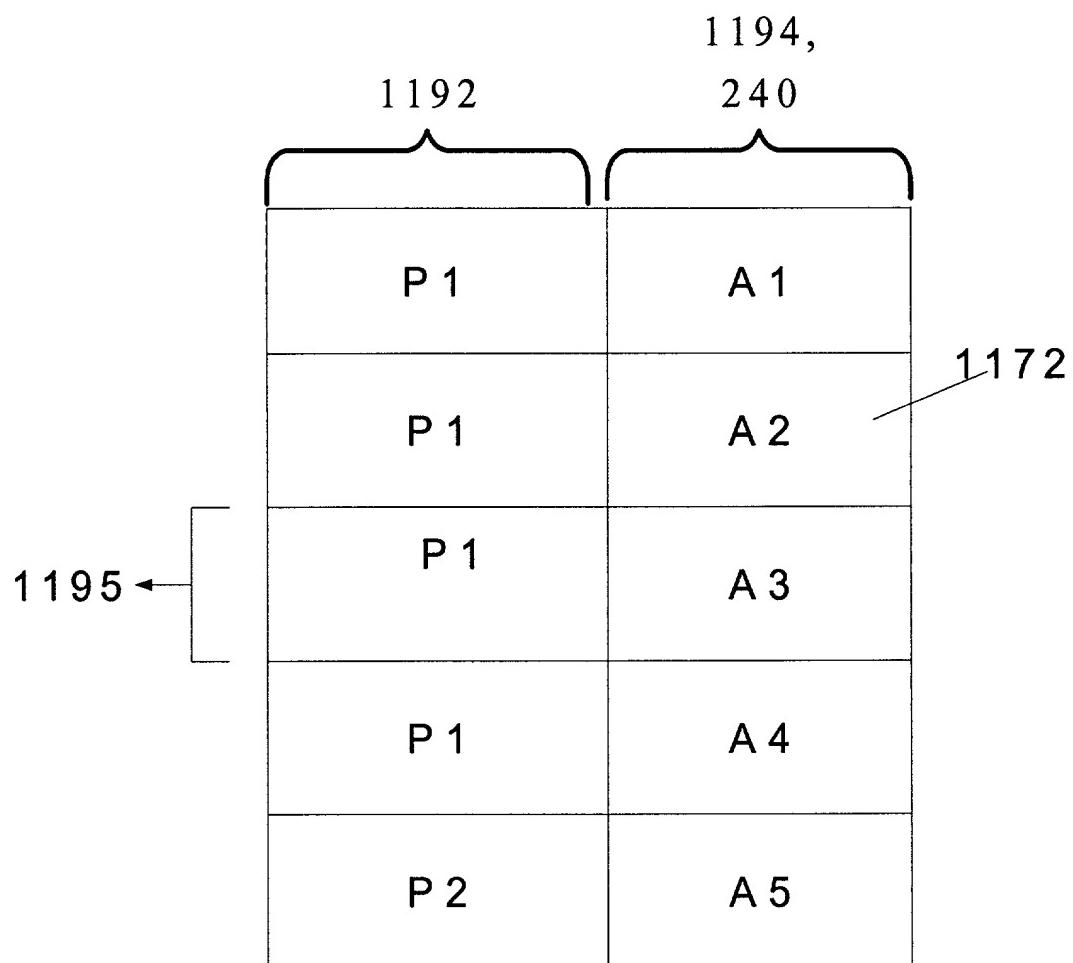


Figure 2C

1190



Package Content Data  
Structure

Figure 3

1170

Asset ID	Location (Machine Location URL)	Name	Asset Type	Other (Optional)
1172	1174	1176	1178	1179
A 1			240T	1175
A 2				
A 3				

1194

Asset  
Definition Data  
Structure

**Figure 4**

Deployable Asset      1370  
Data Structure

Asset ID	Version
1372	1374

1375

1380

Client  
Deployment  
Queue

Target/ Client ID
1382

1385

Figure 5

Figure 7

1390

Cient ID	Client Assets
1392	1394

Client Asset  
Table

Figure 6

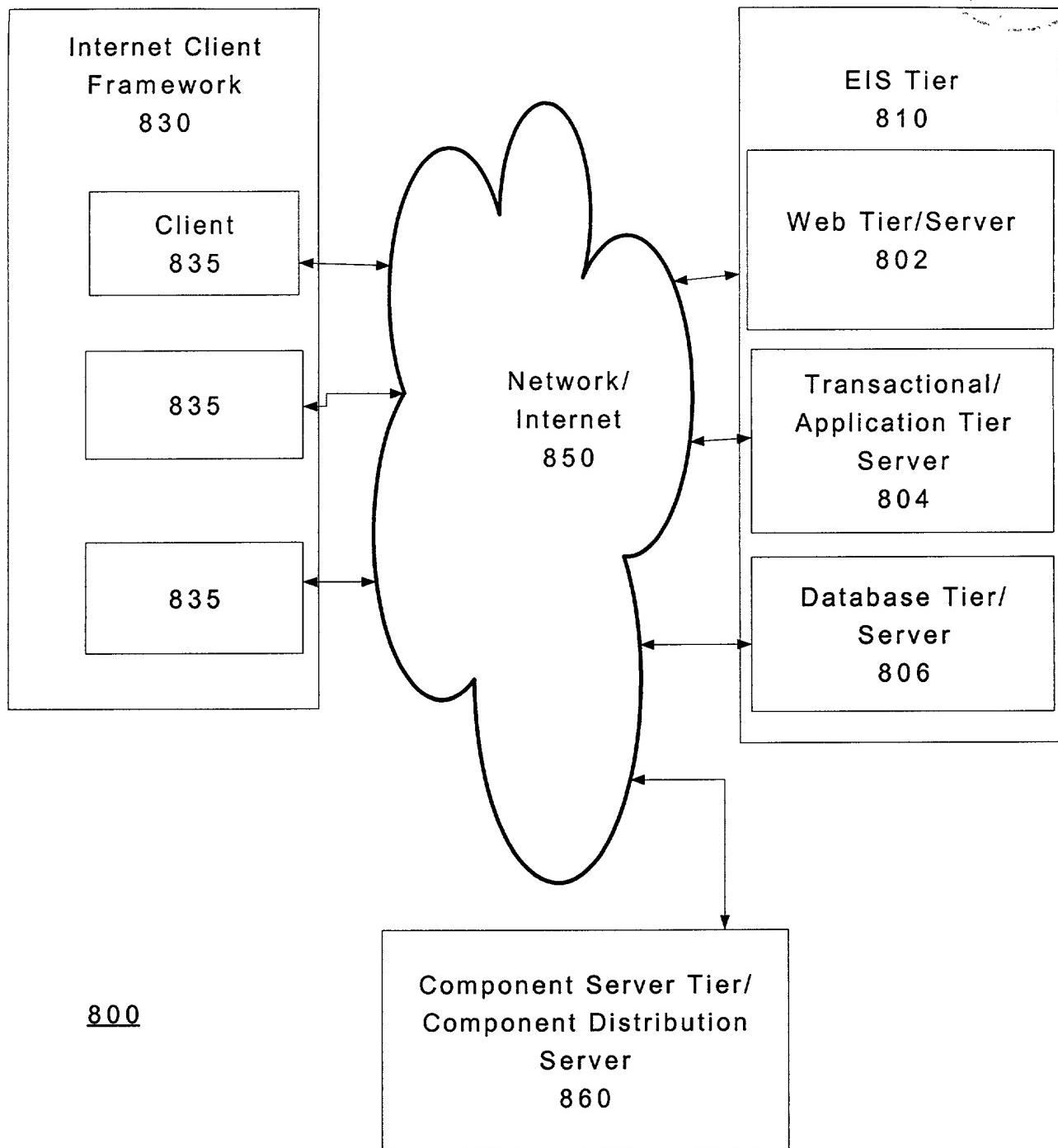


FIGURE 8

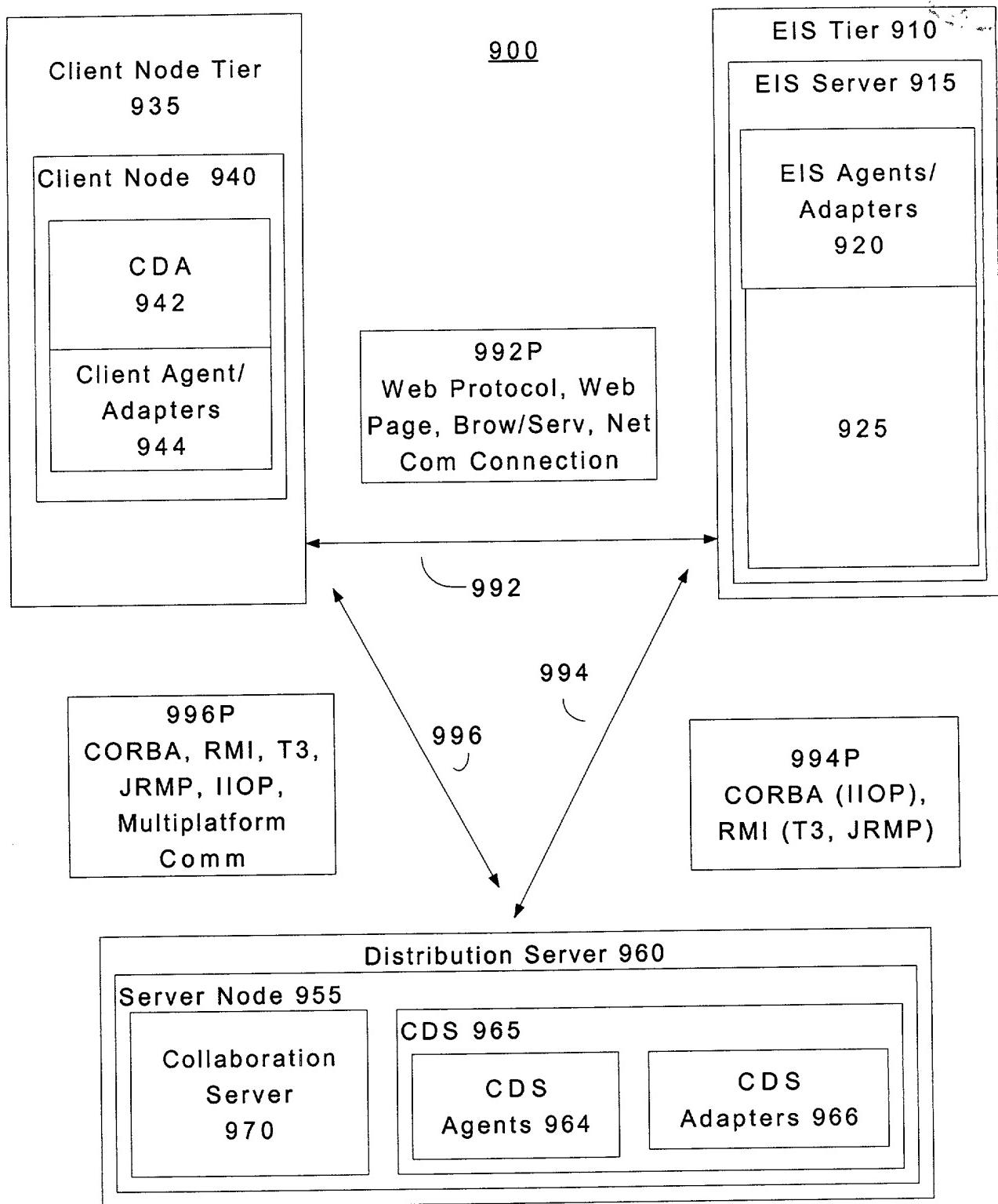


FIGURE 9

1000

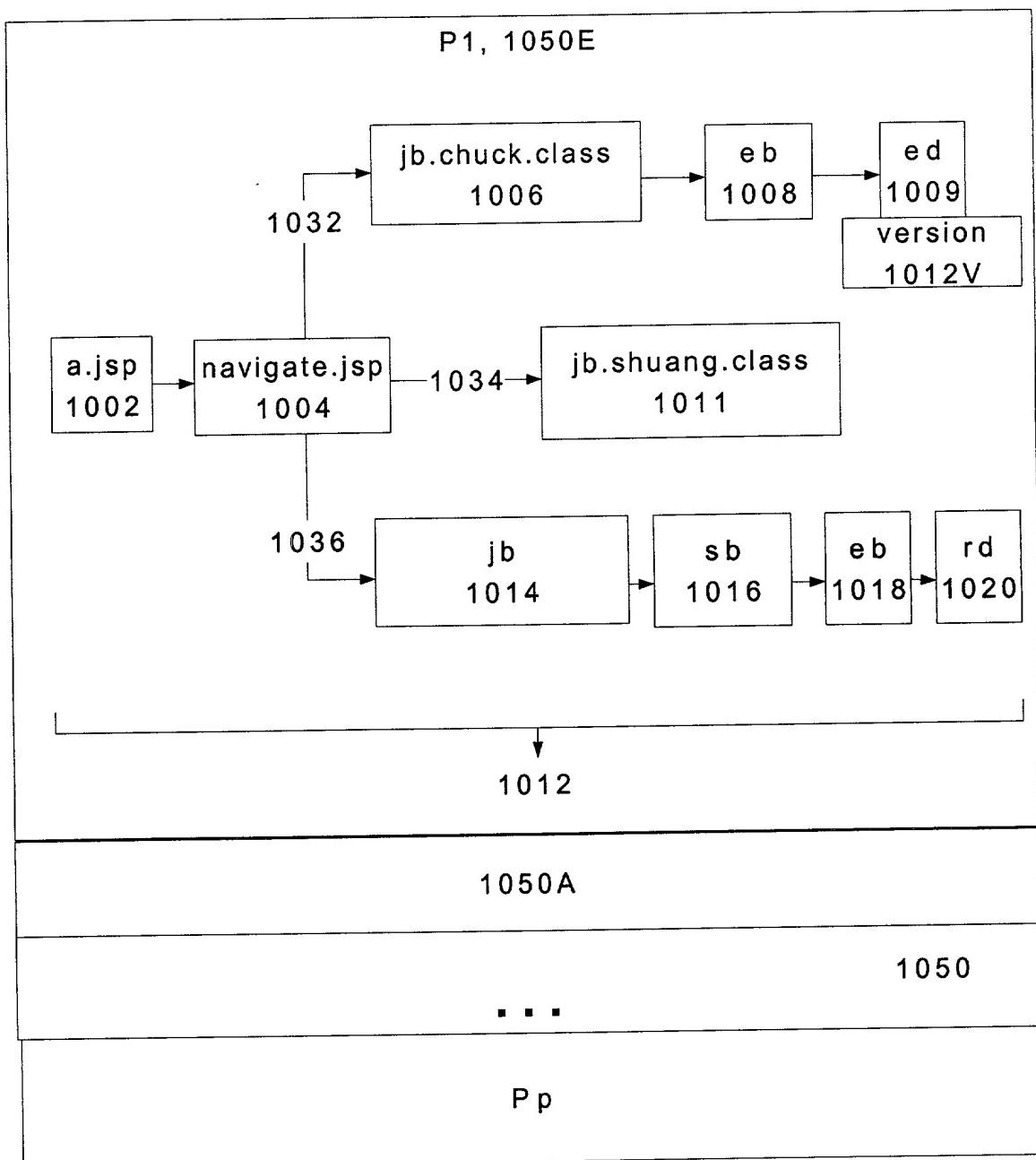


Figure 10

1100

1150		1160		1162		1163	
Package ID	Immediate	Delivery Start Time	Delivery End Time	Expire Time	Remove Time	Refresh Time	Location (e.g URL)
1110	1152	1154	1156	1158	1160	1162	
1105	P1						
	P1						
	P1						

1105

Package Definition Data Structure

Figure 11

1100A

Extended Environment-  
Package (EEP)  
1120A

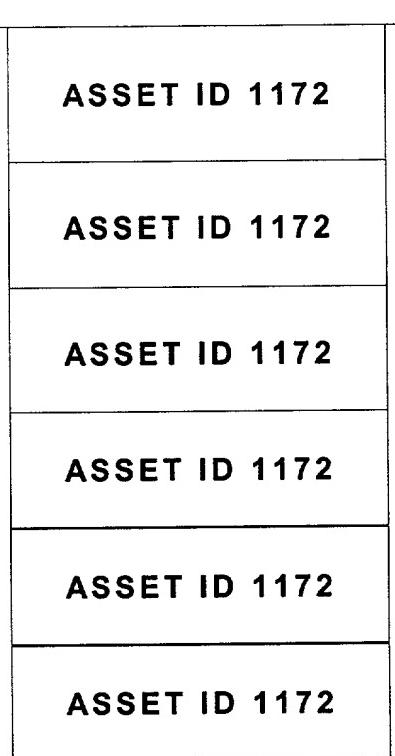
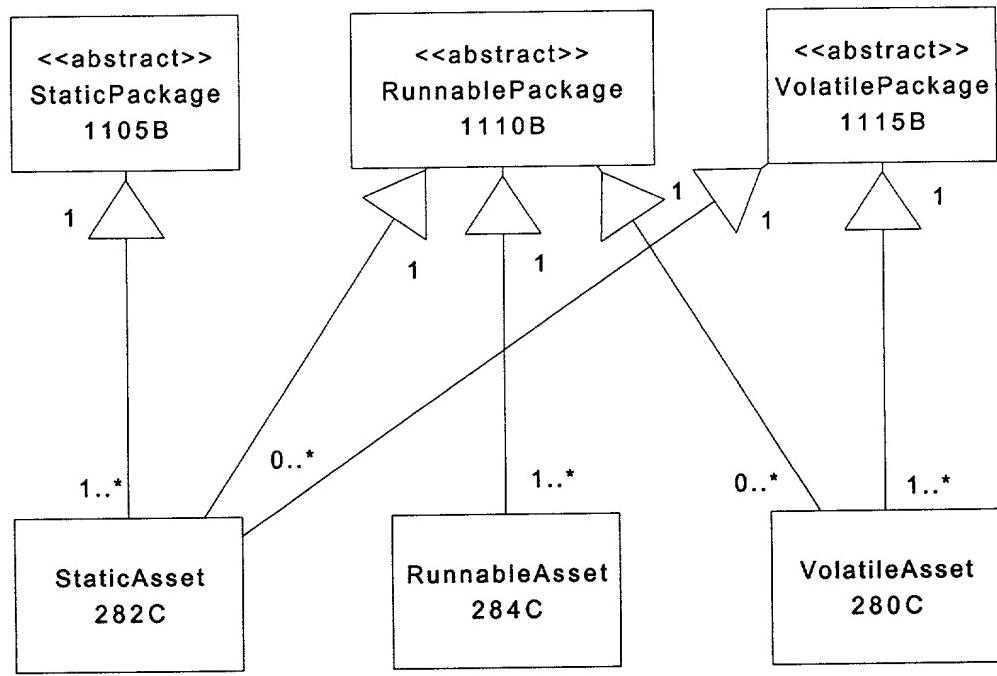


FIGURE 11A



1100B

Figure 11B

1120A

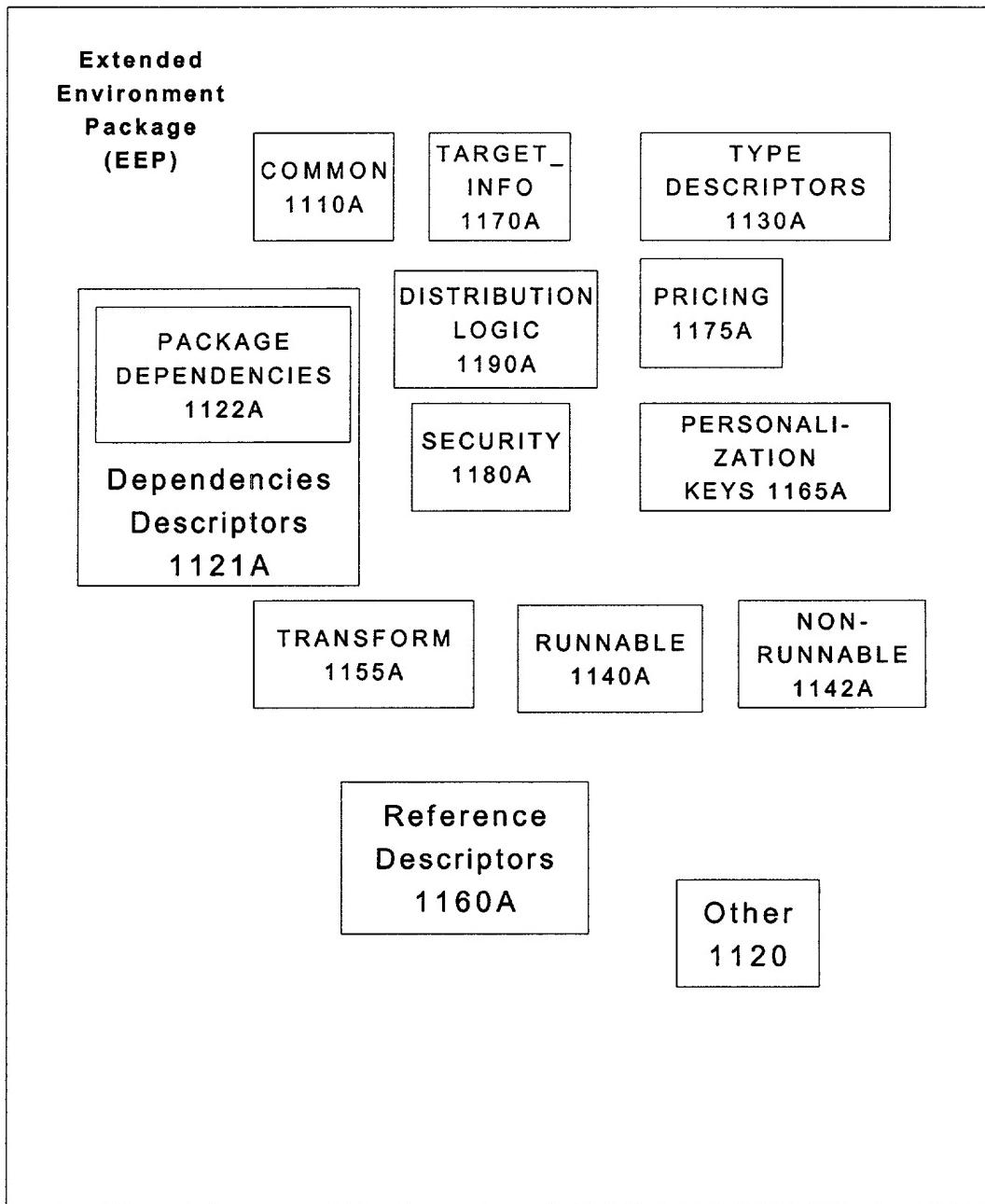
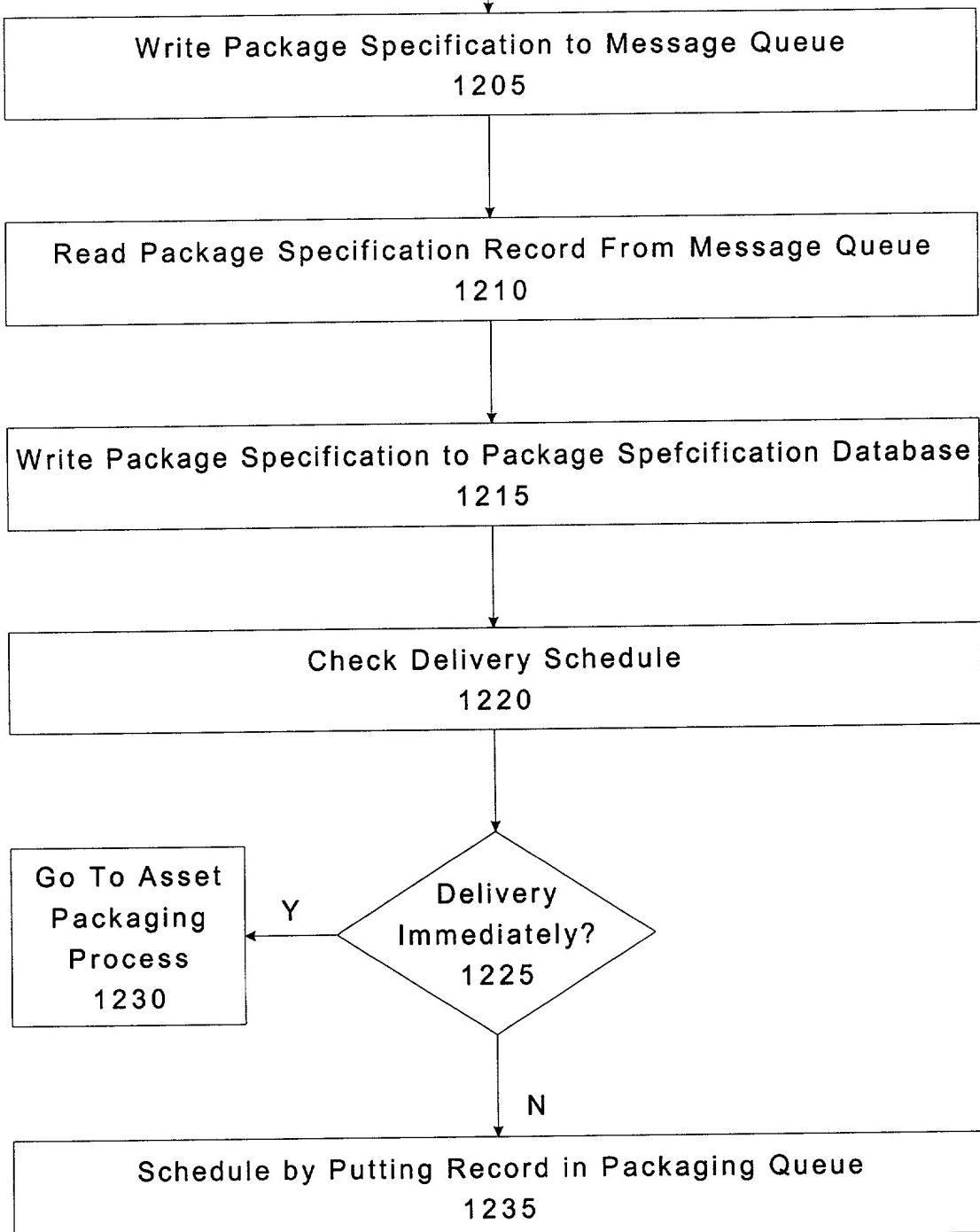


FIGURE 11C

1200



### Package Specification Process

Figure 12

1250

1255

Package ID 1252	Start Time 1254

Packaging  
Queue

Figure 12A

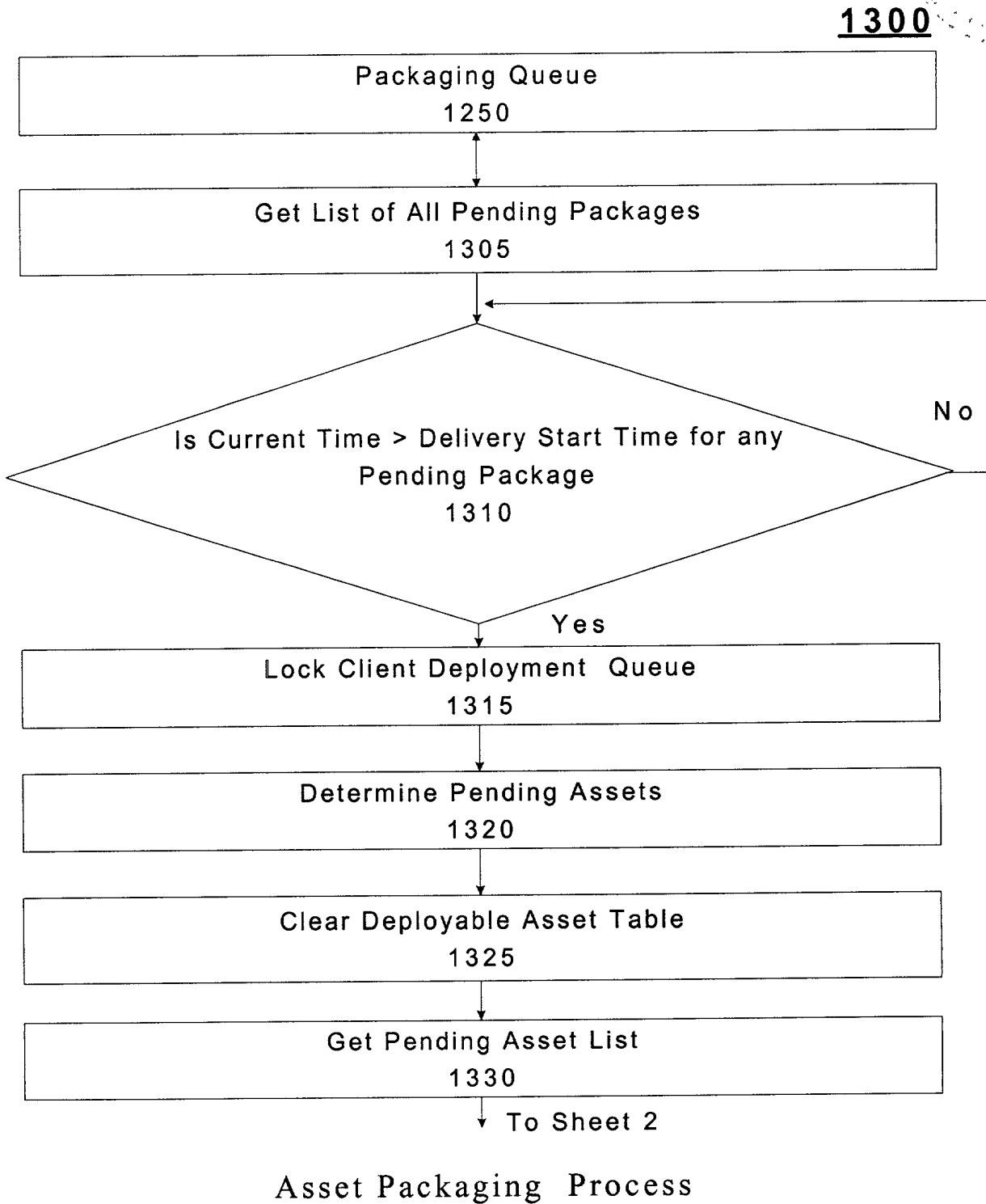
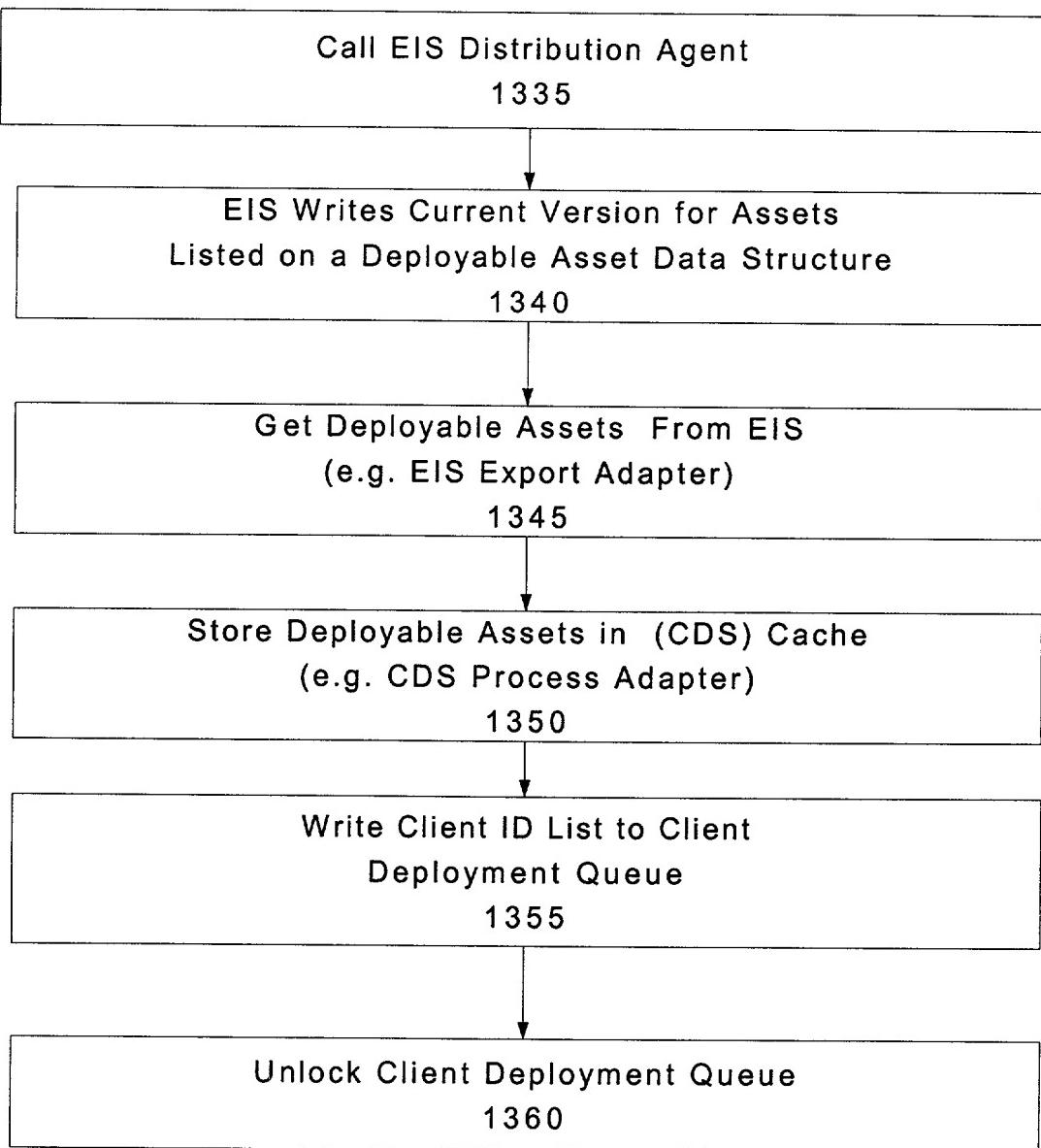


Figure 13 - Sheet 1

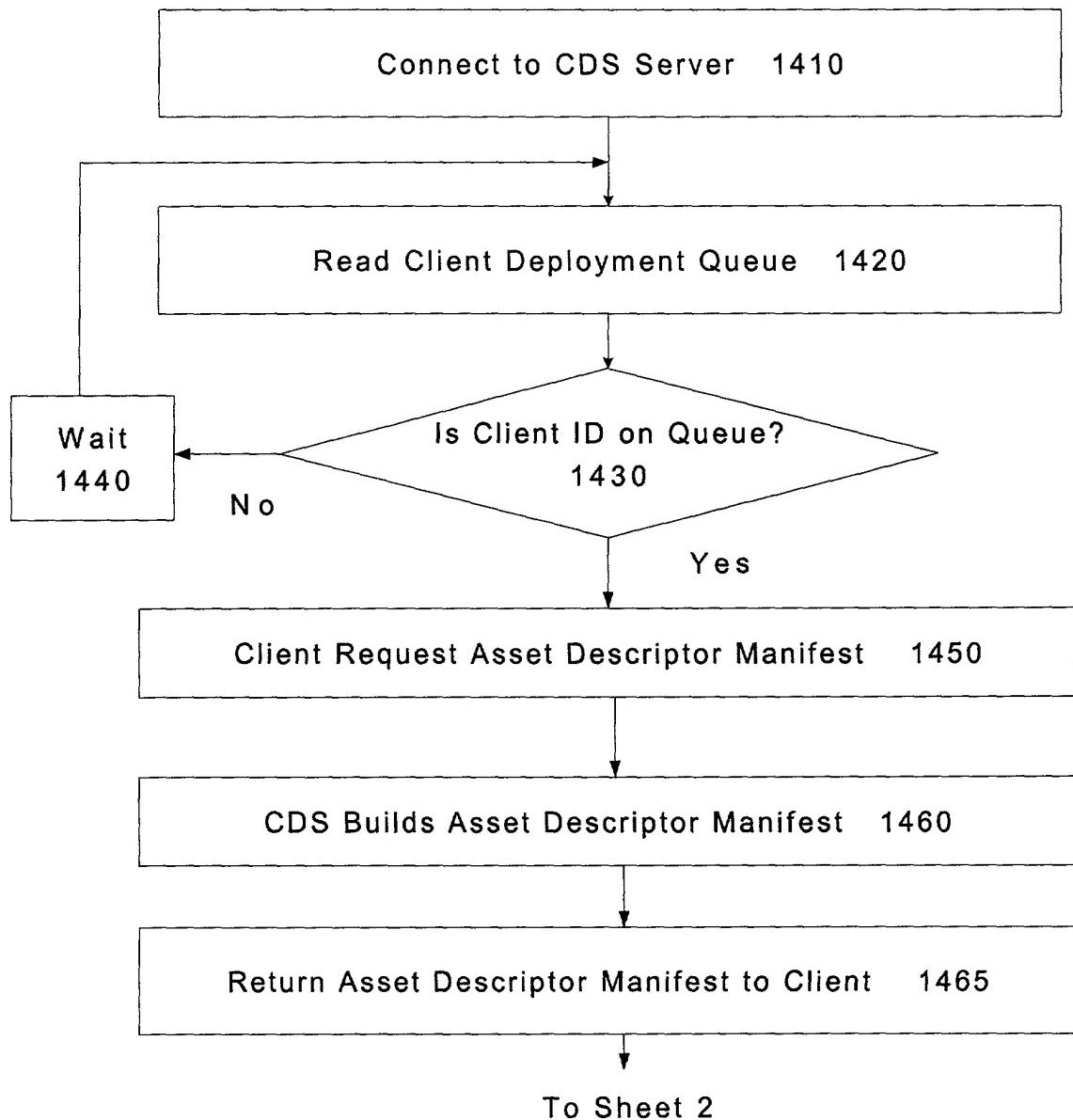
1300

From Sheet 1



## Asset Packaging Process

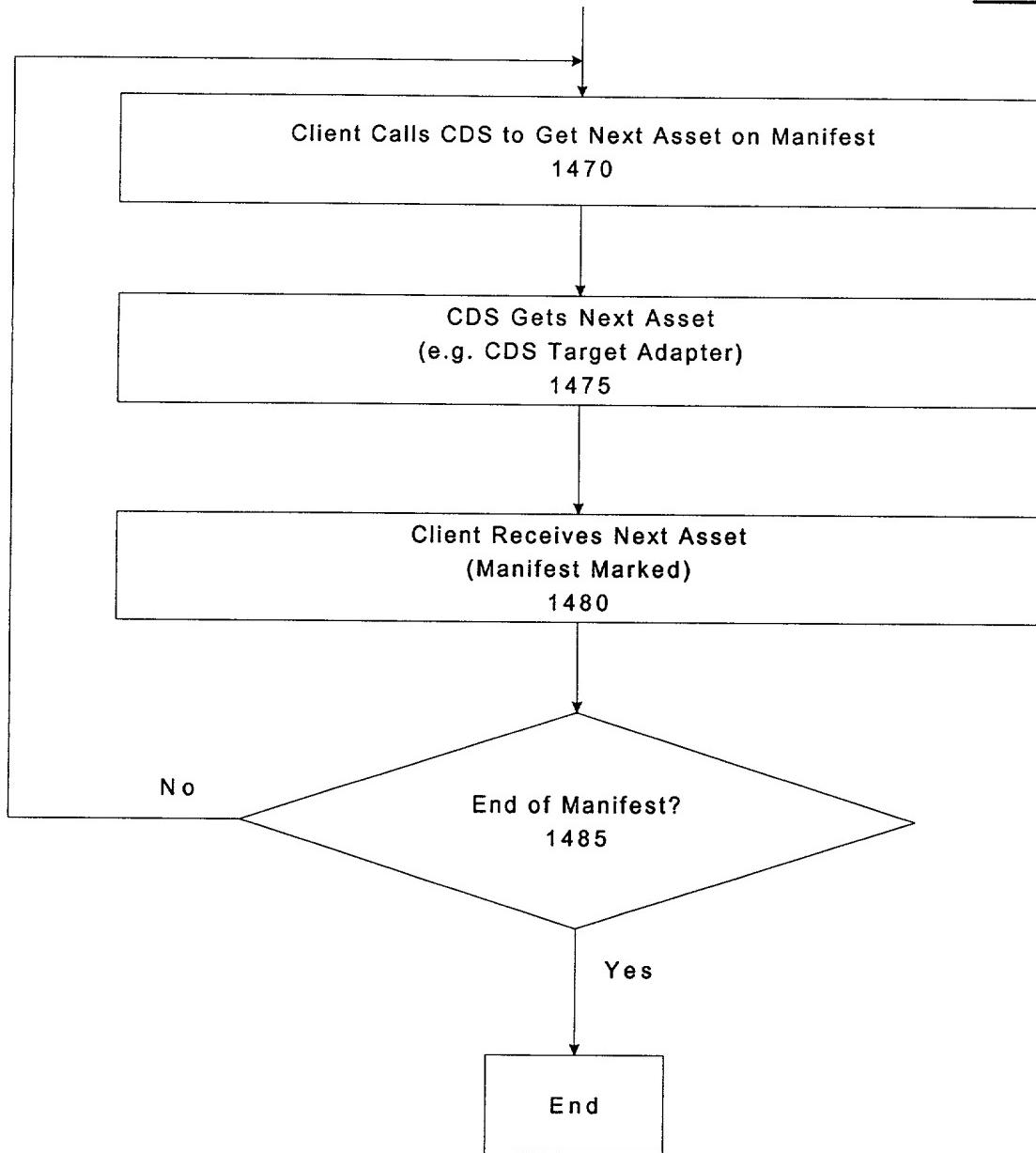
Figure 13 - Sheet 2

1400

## Client Deployment Process

Figure 14 - Sheet 1

From Sheet 1

1400

Client Deployment Process

Figure 14 - Sheet 2

1490

Client ID				
Asset ID	Offset	Asset Type	Cache Name	Version
1454	1456	1458 (Optional)	1478	1479

1453

Asset Descriptor  
Manifest Data  
Structure

Figure 14 A

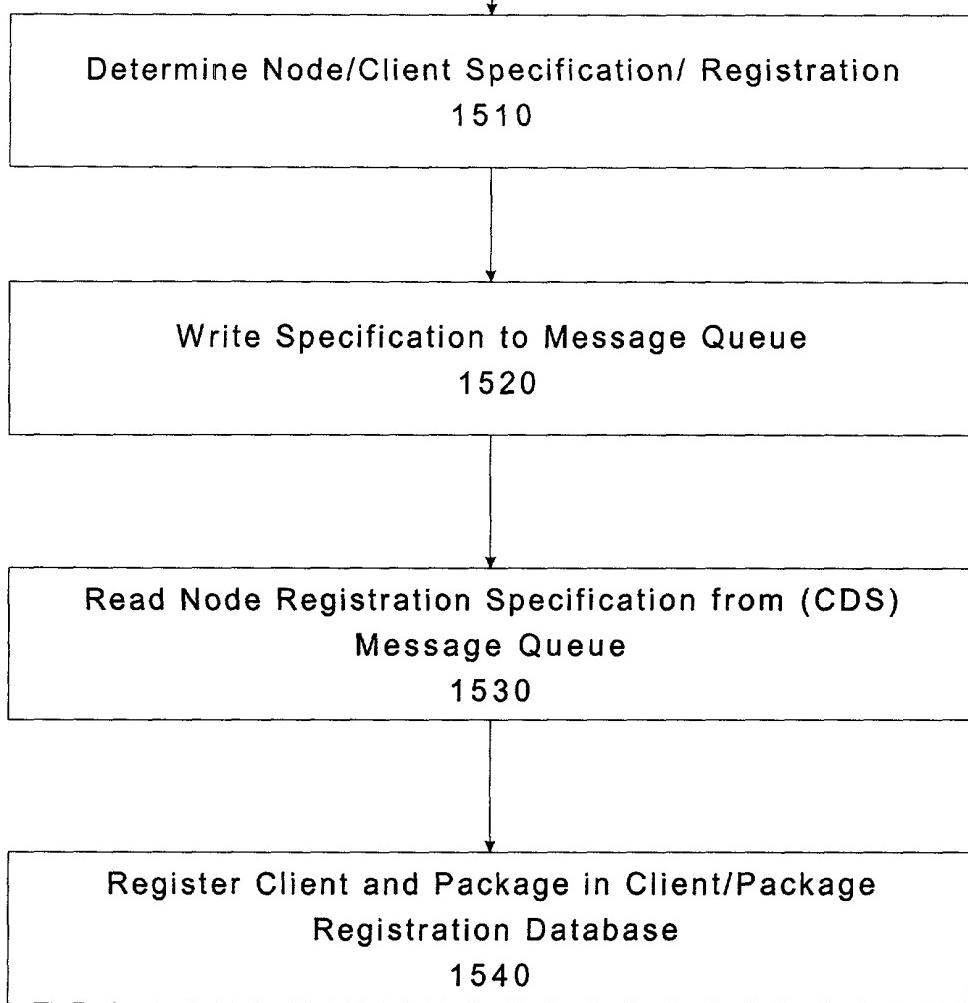
1495

Client ID 1452	Asset ID 1454	Version (Timestamp) 1479
-------------------	------------------	--------------------------------

Client Asset Table

Figure 14B

1500



### Node Registration Process

Figure 15

1522

Node ID 1524	Package ID 1526

**Node Registration Specification**

Figure 15A

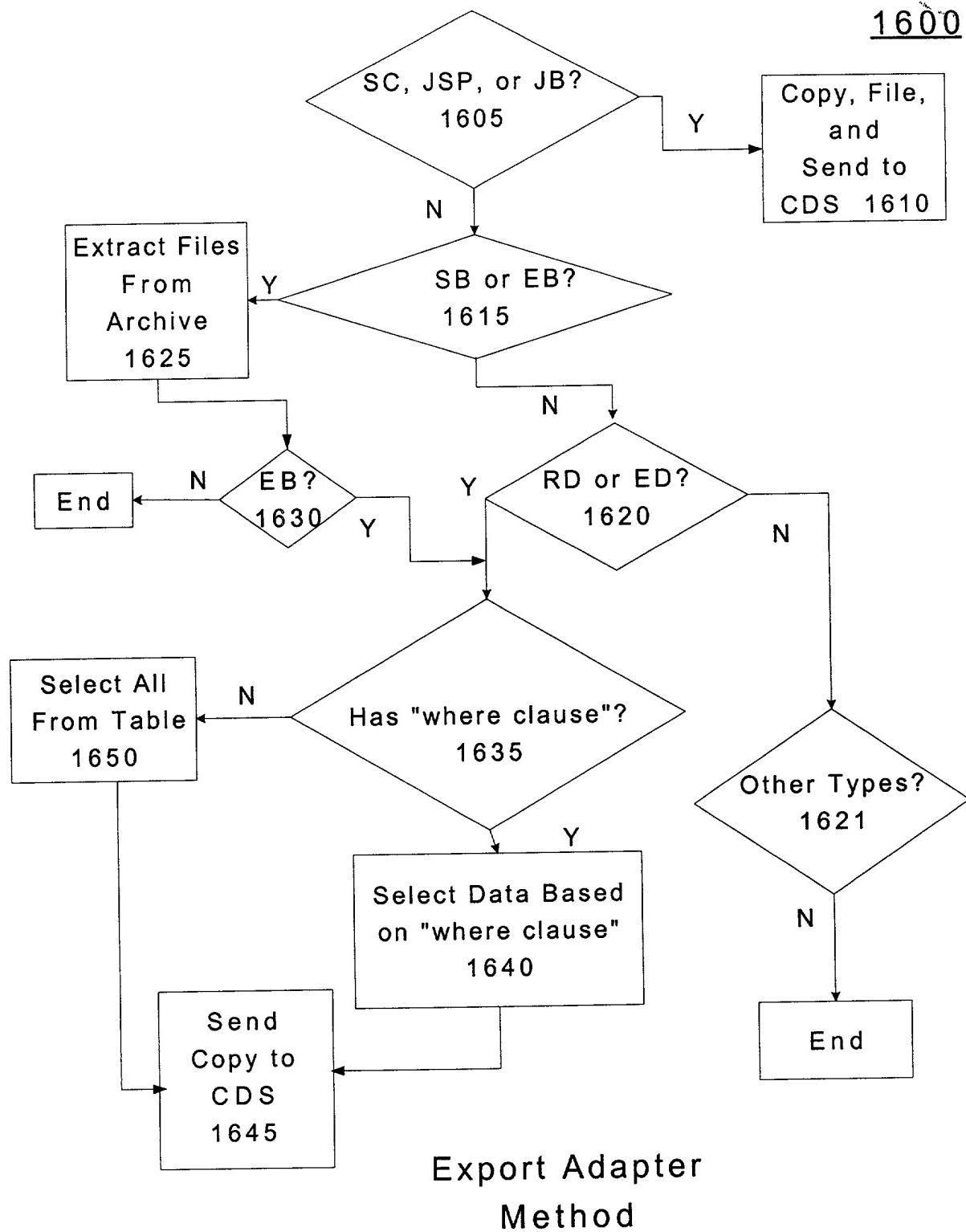
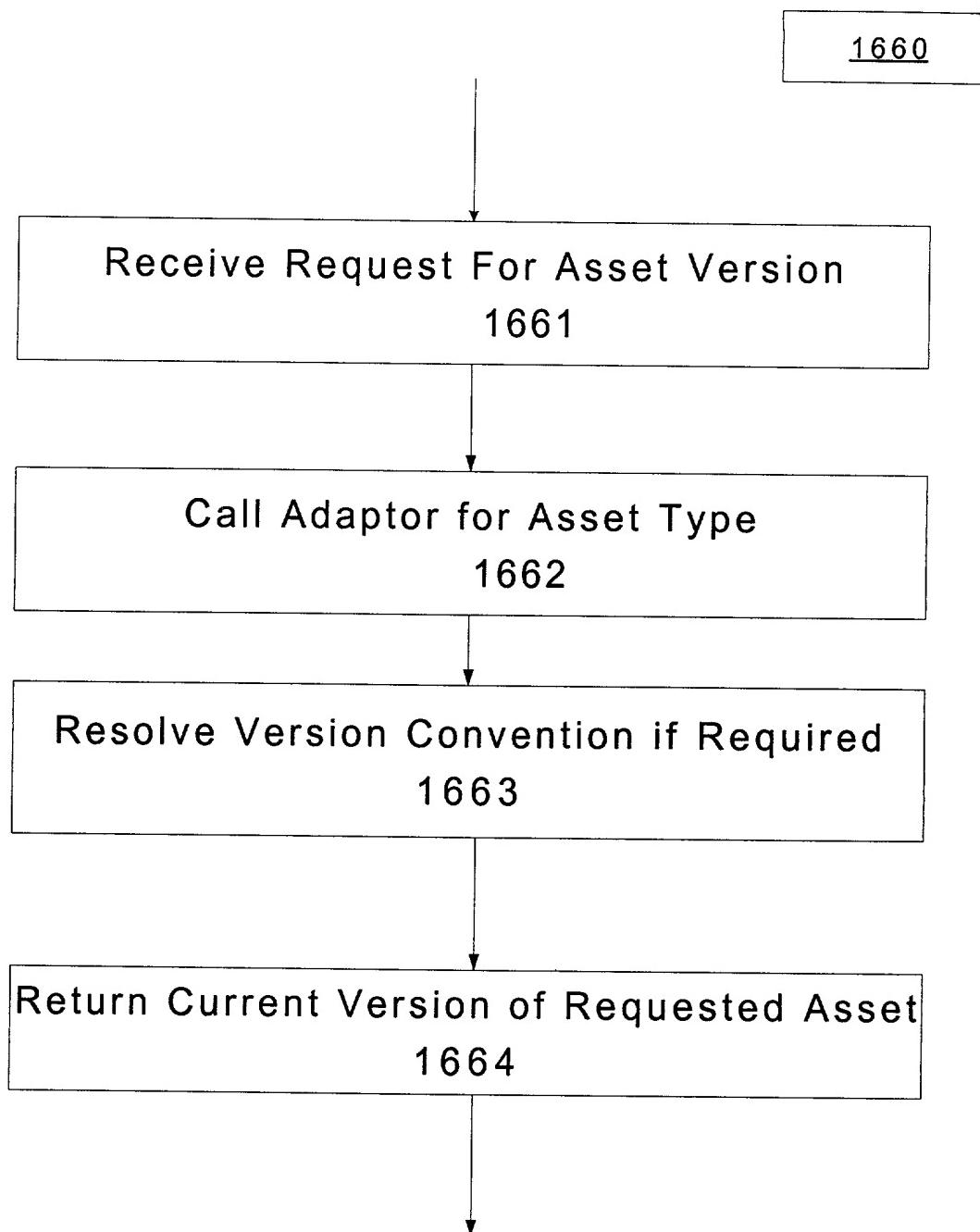


Figure 16



Version Asset Adapter Process - VAM

Figure 16A

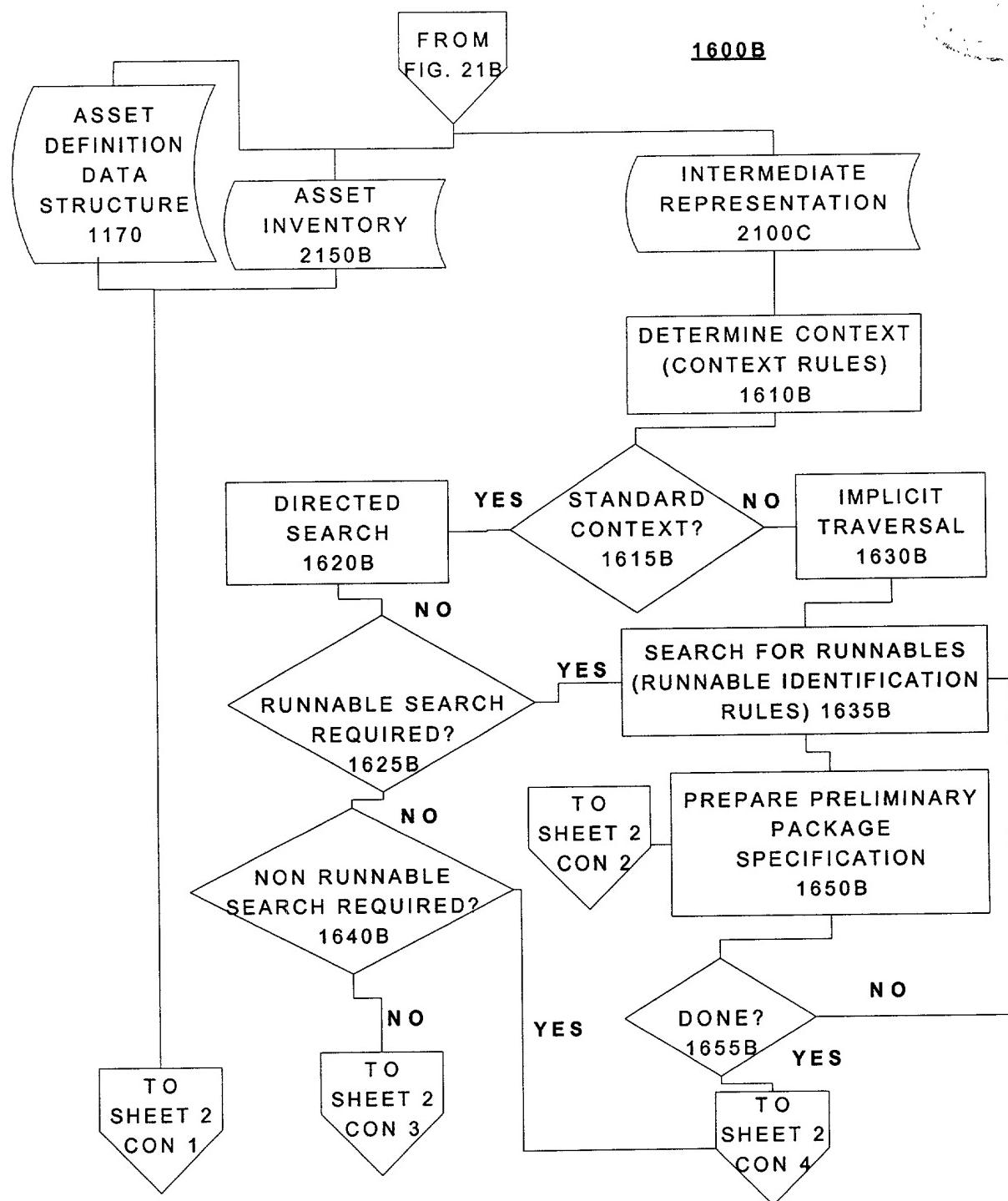


FIG. 16B  
SHEET 1

1600B - 2

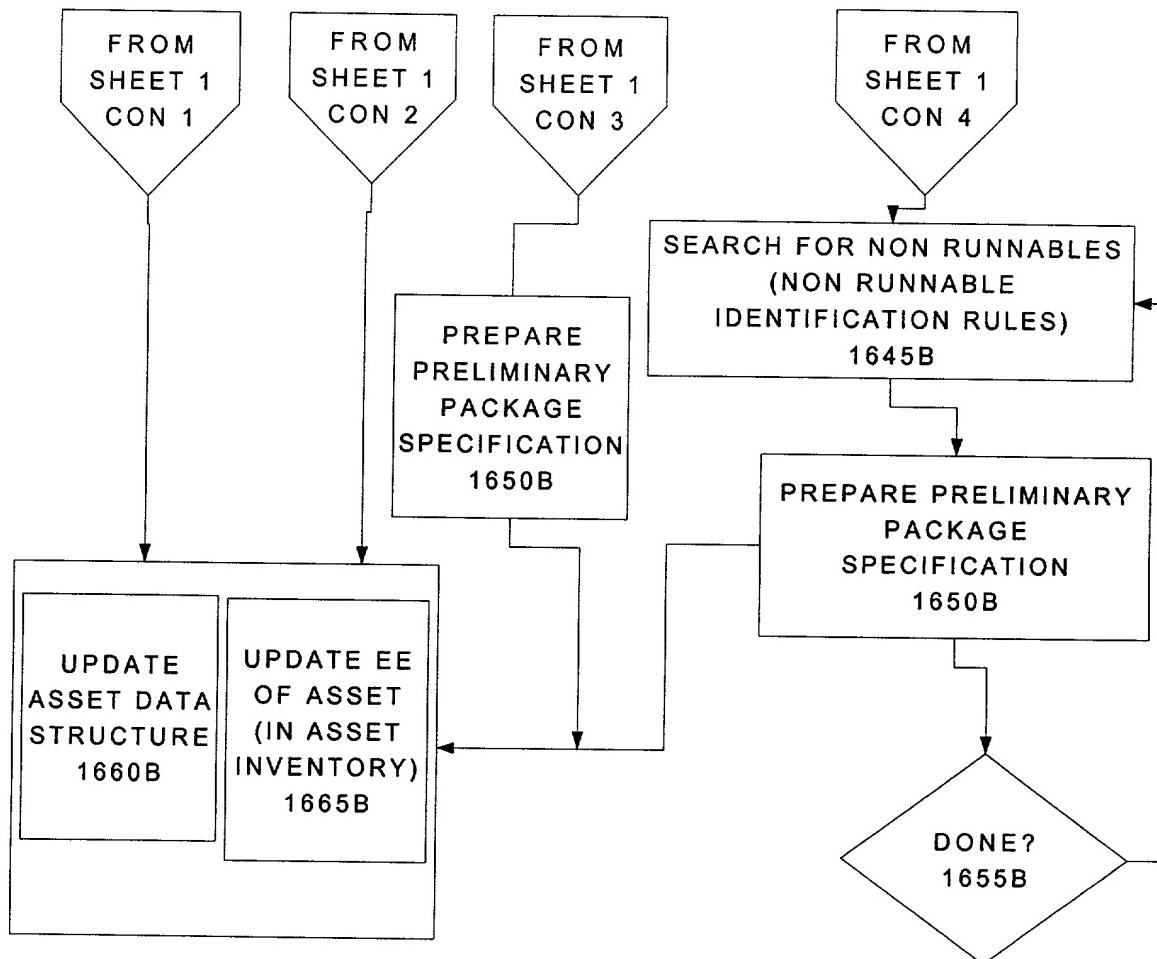
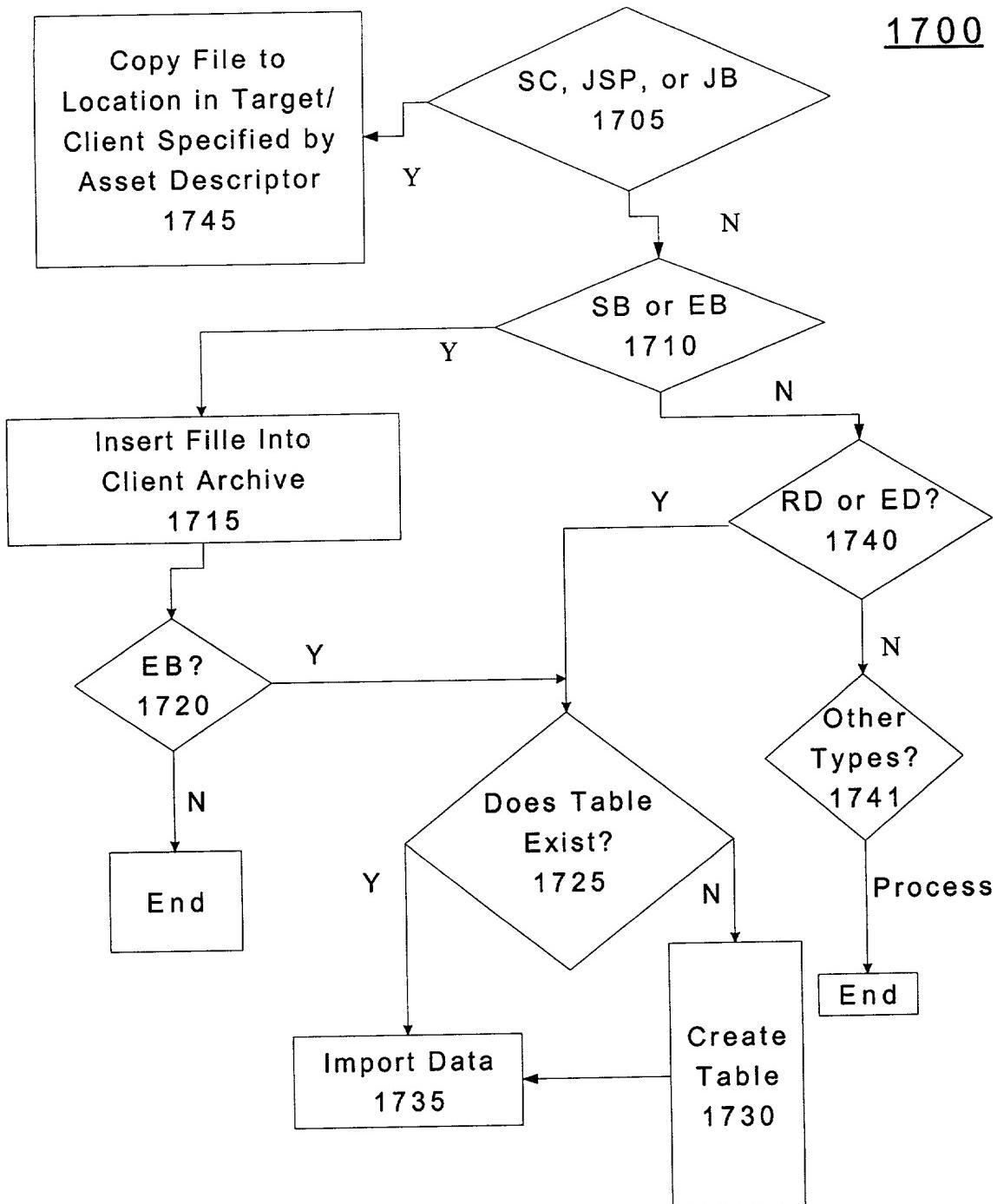


FIG. 16B  
SHEET 2



### Deploy Adapter Method

Figure 17

1700A

DIS TRANSACTIONAL DEPLOYMENT SPHERE OF CONTROL

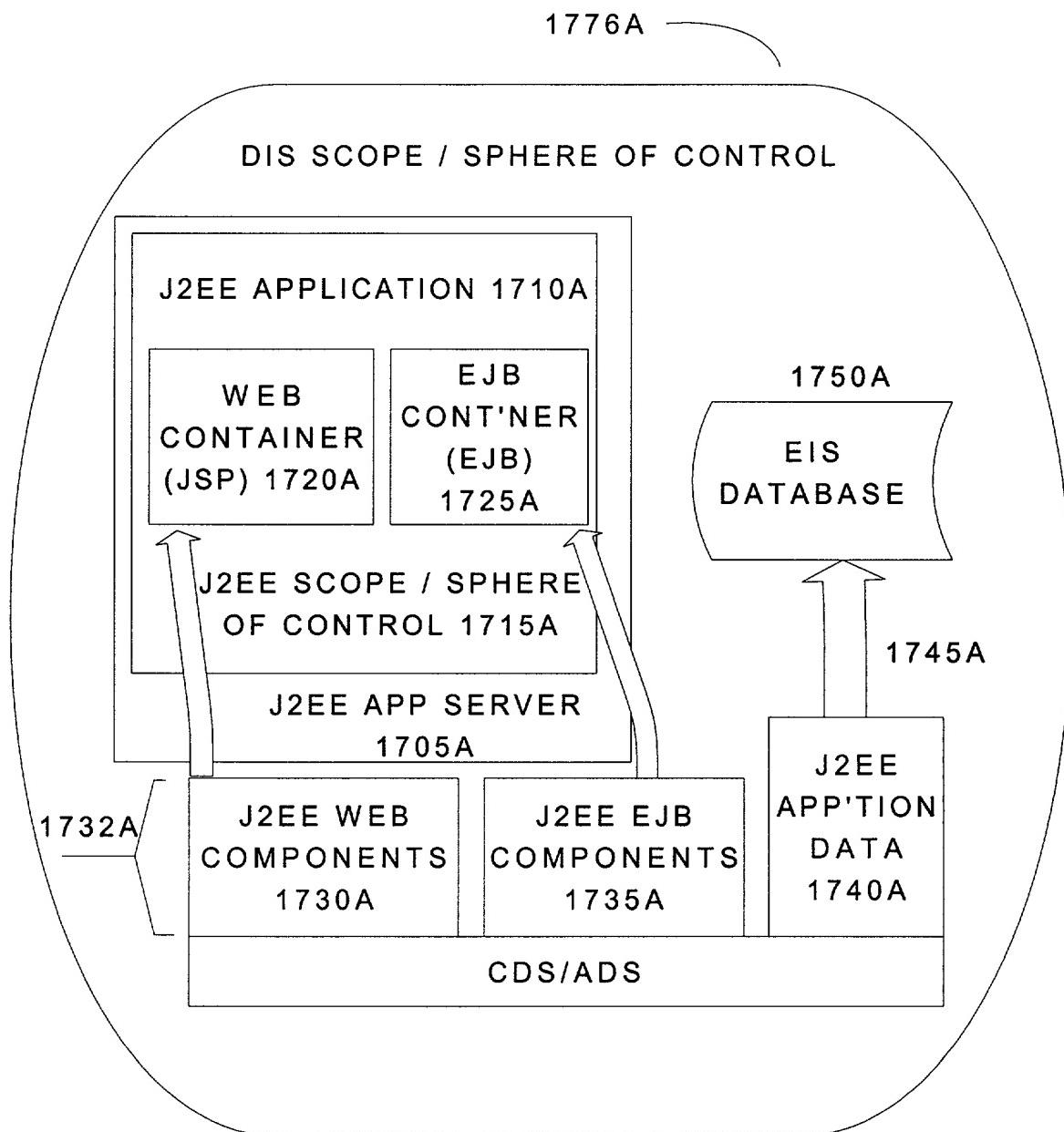


FIGURE 17A

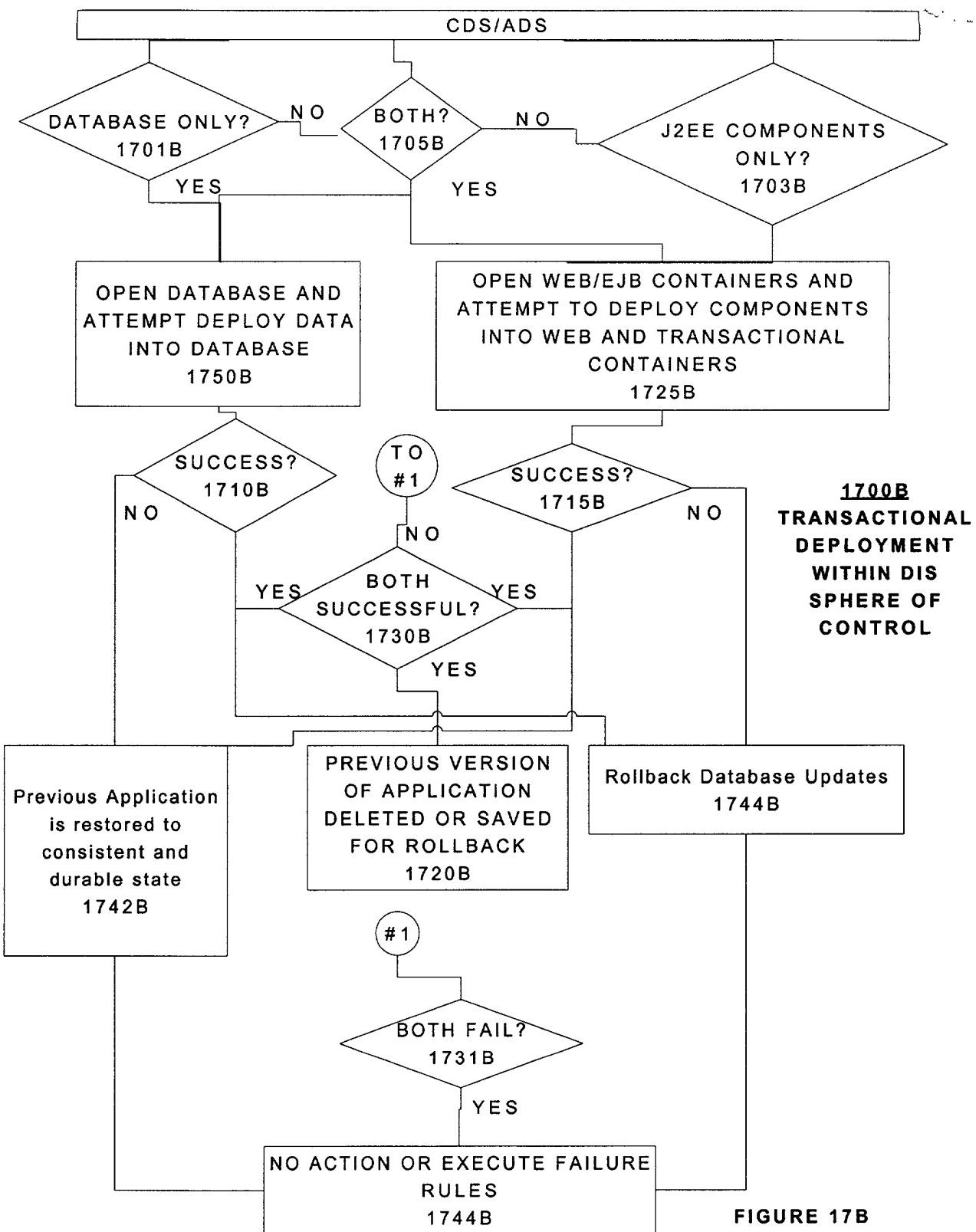
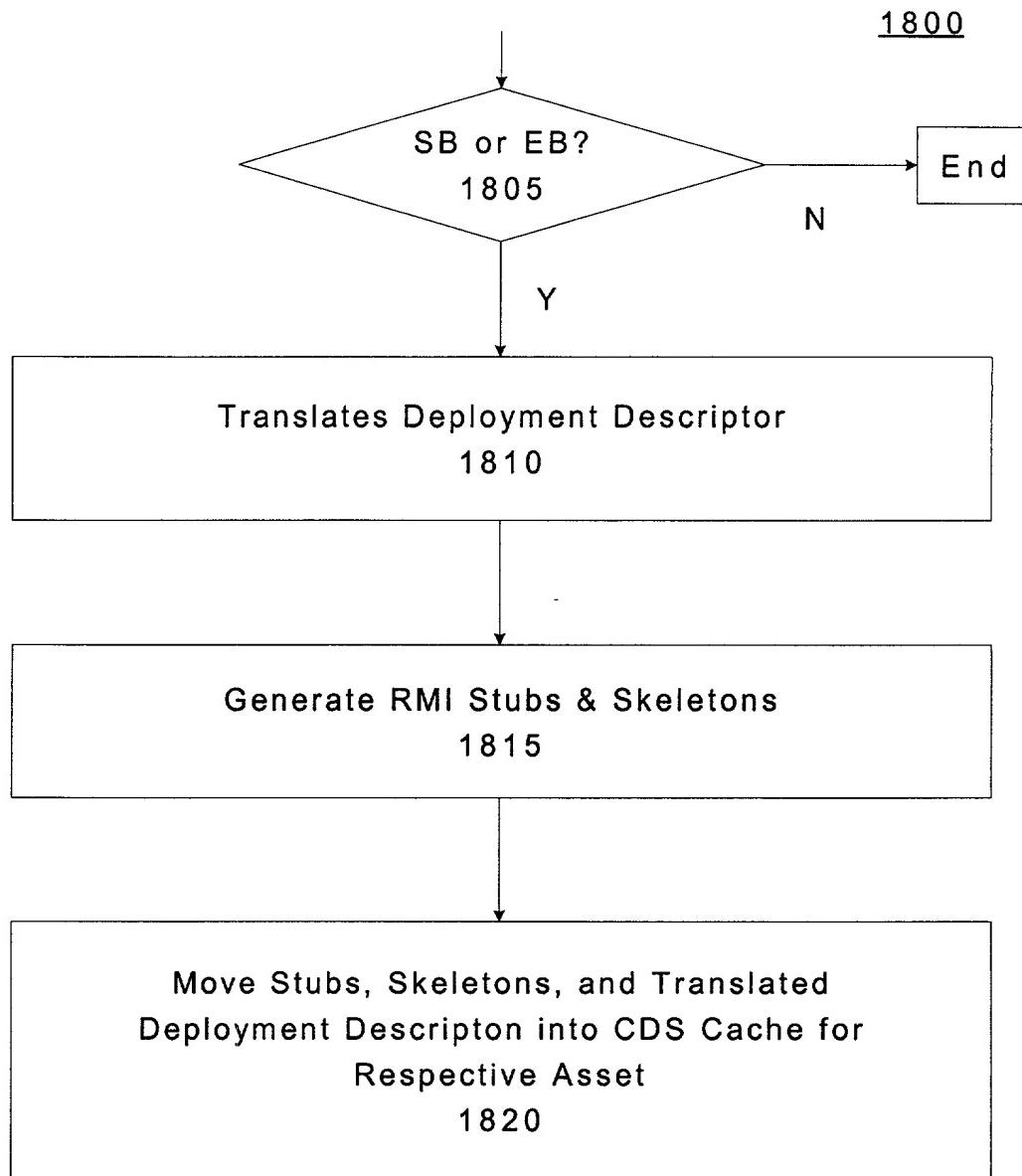


FIGURE 17B



Process Adaptor Method

Figure 18

1800A

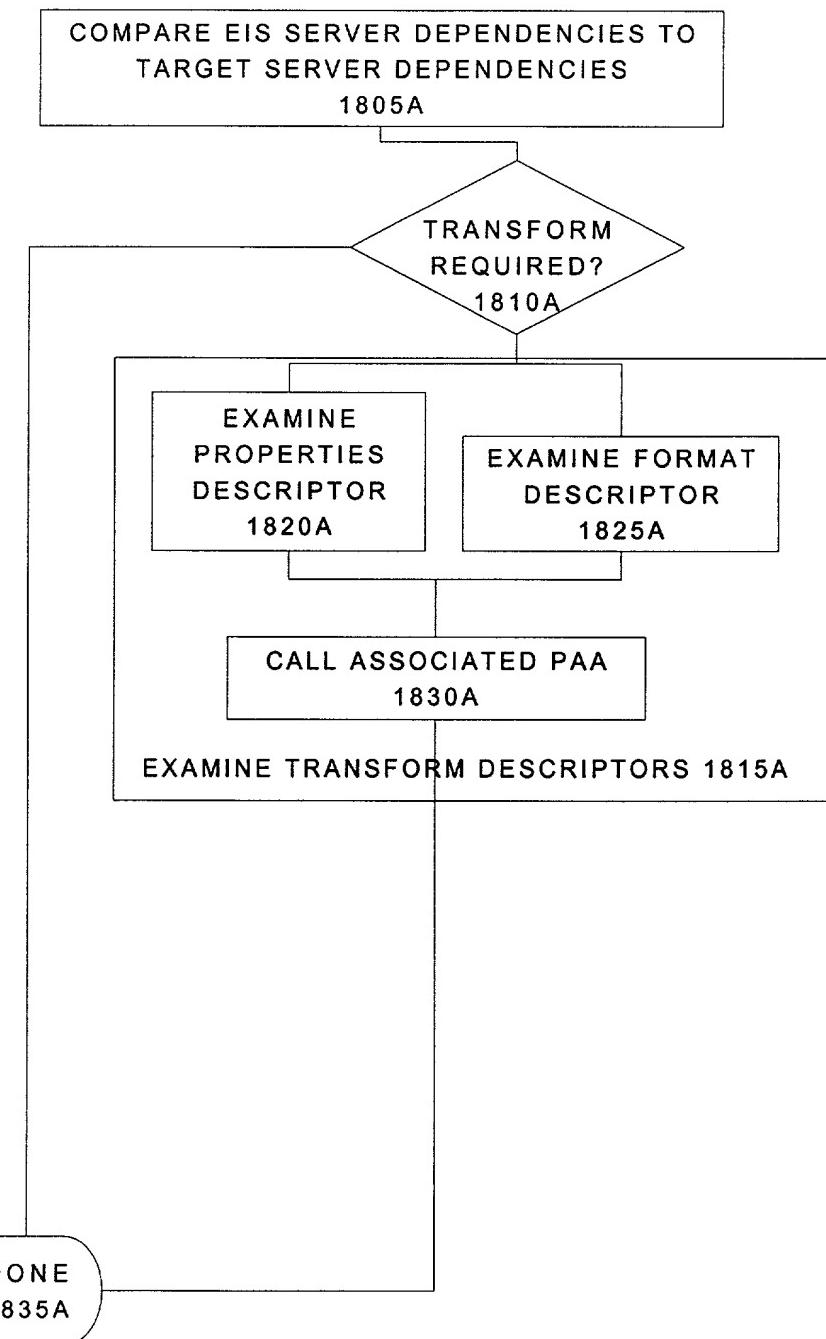
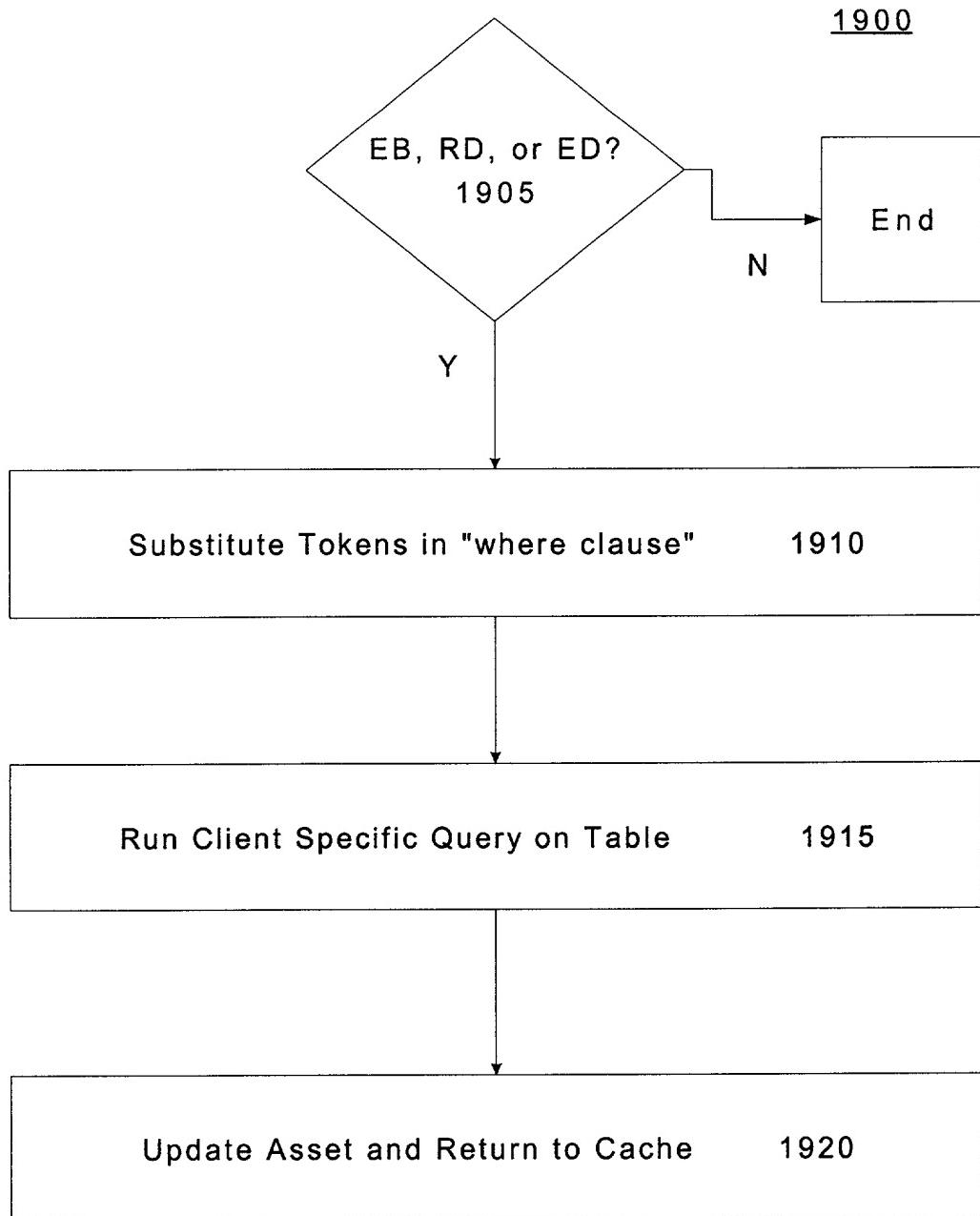


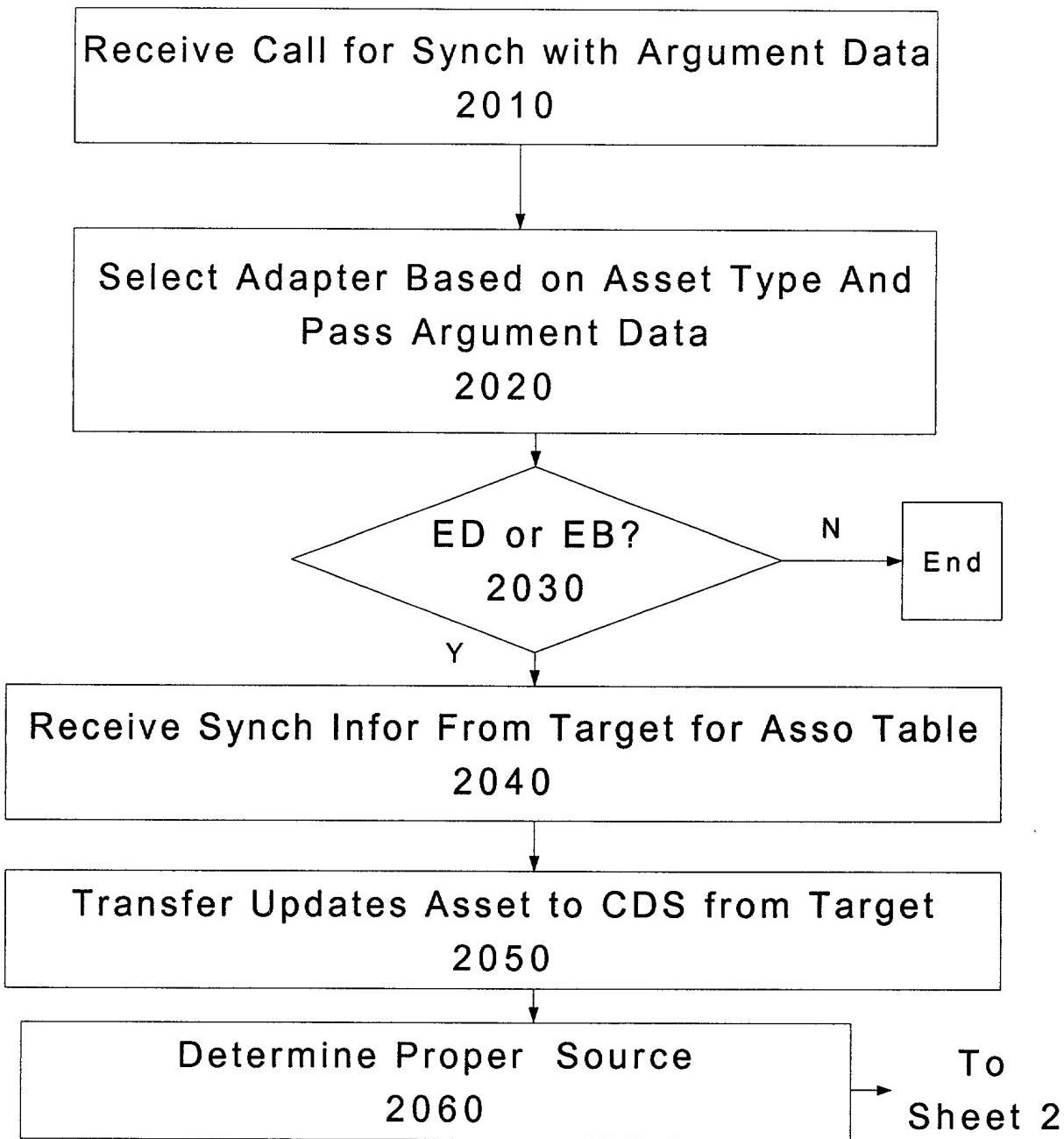
FIGURE 18A



Target Adapter Method

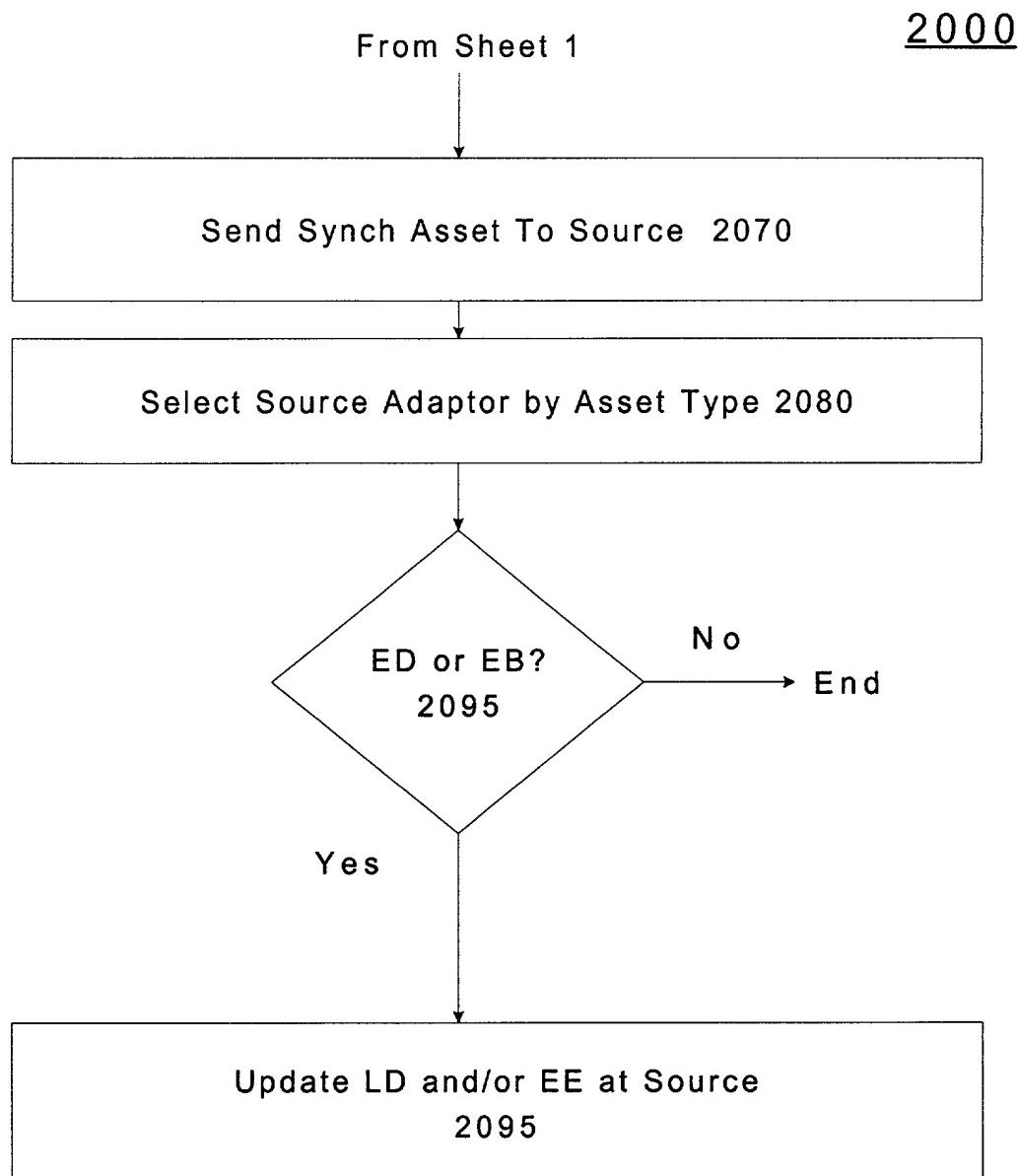
Figure 19

2000



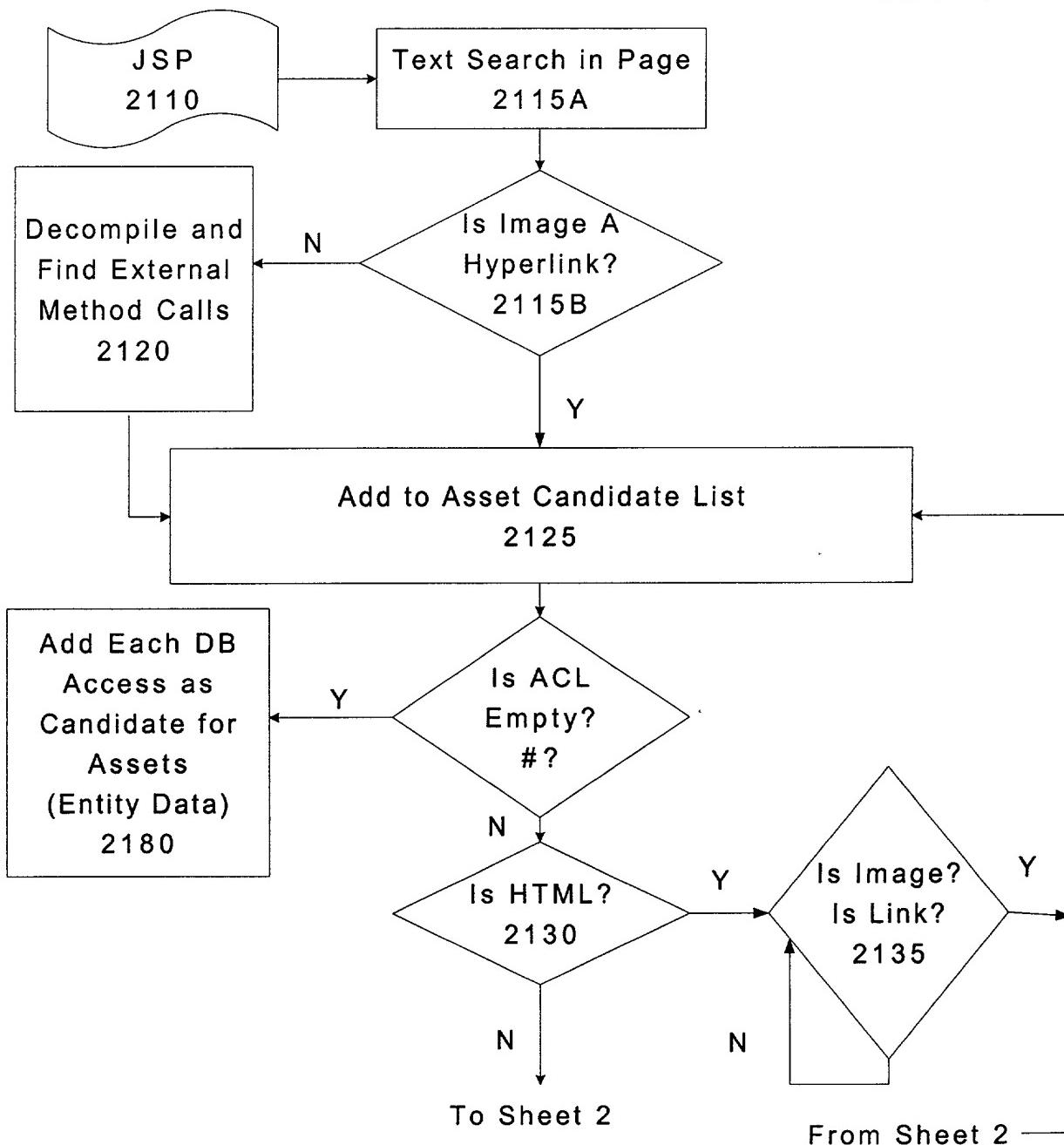
Synchronize Asset Adapter Process

Figure 20 - Sheet 1



Synchronize Asset Adapter Process

Figure 20 - Sheet 2

2100

## Discovery Asset Adapter Method

Figure 21 - Sheet 1

2100

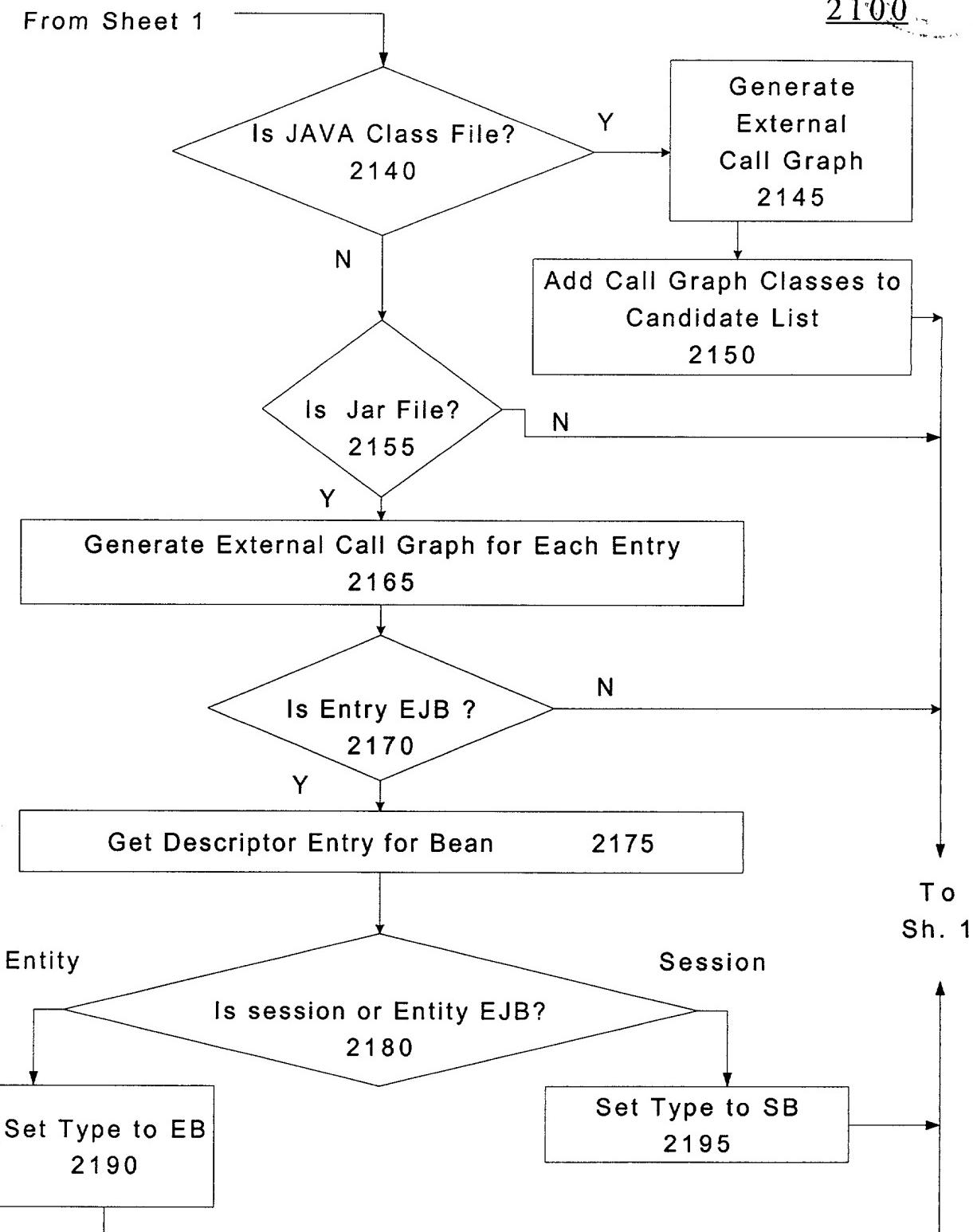
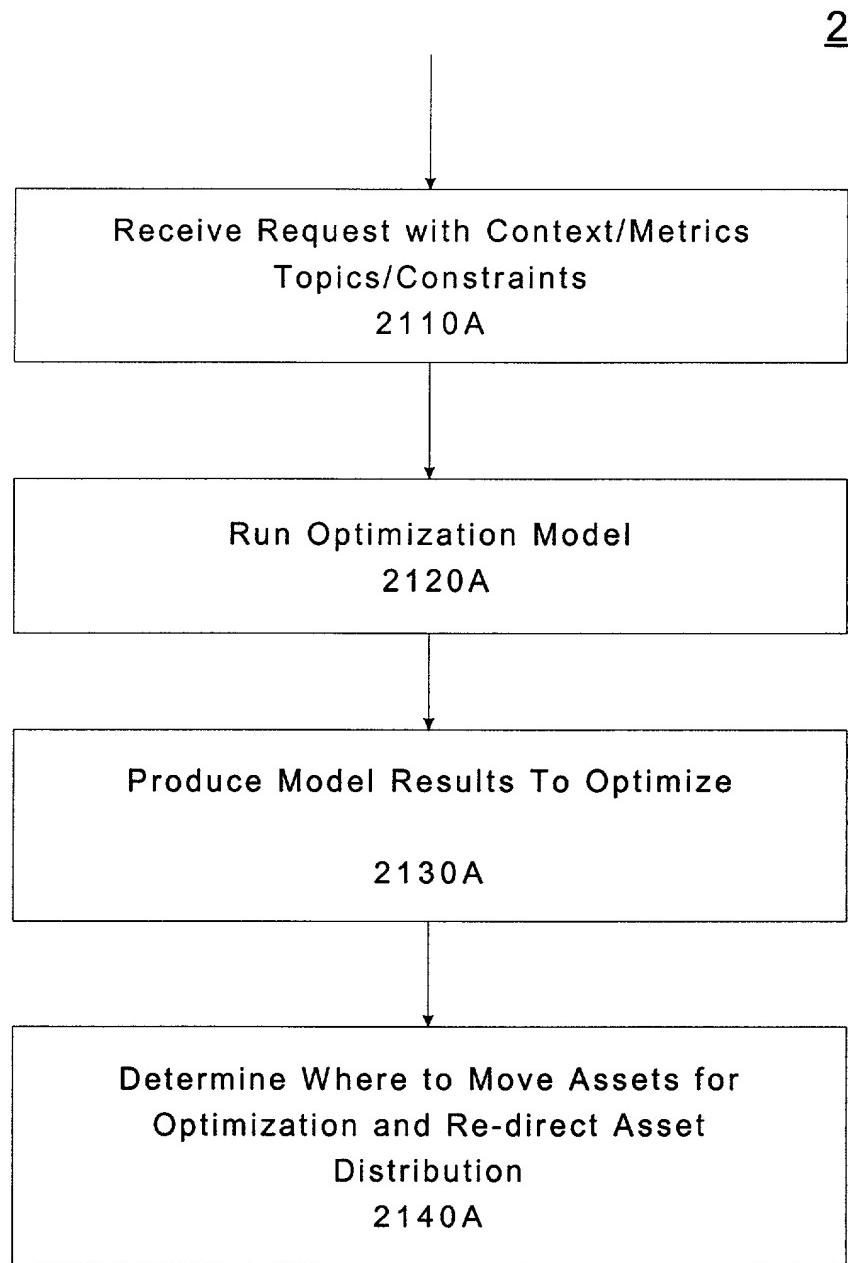


Figure 21-Sheet 2



Adjustment Asset Adapter Process

Figure 21A

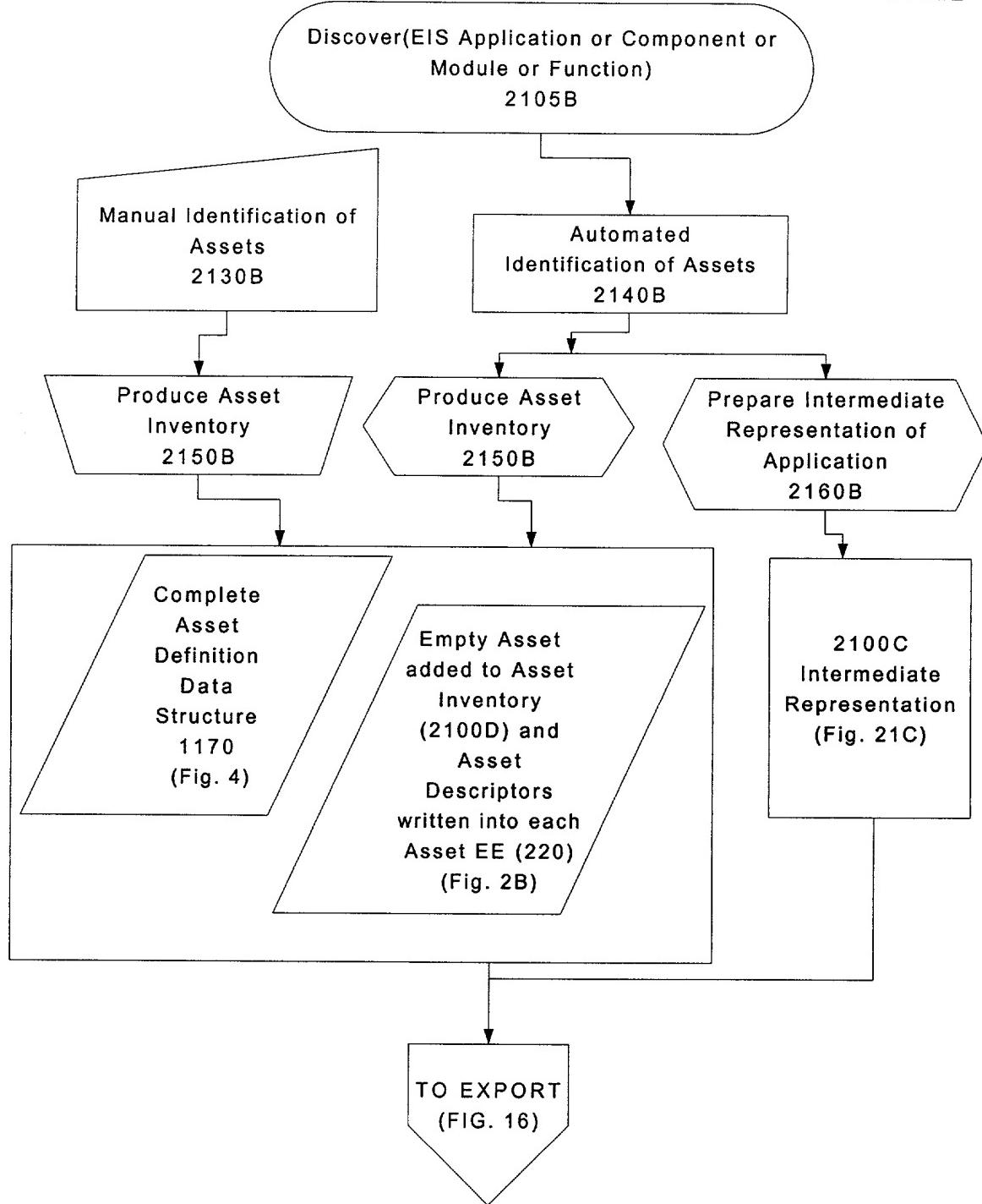
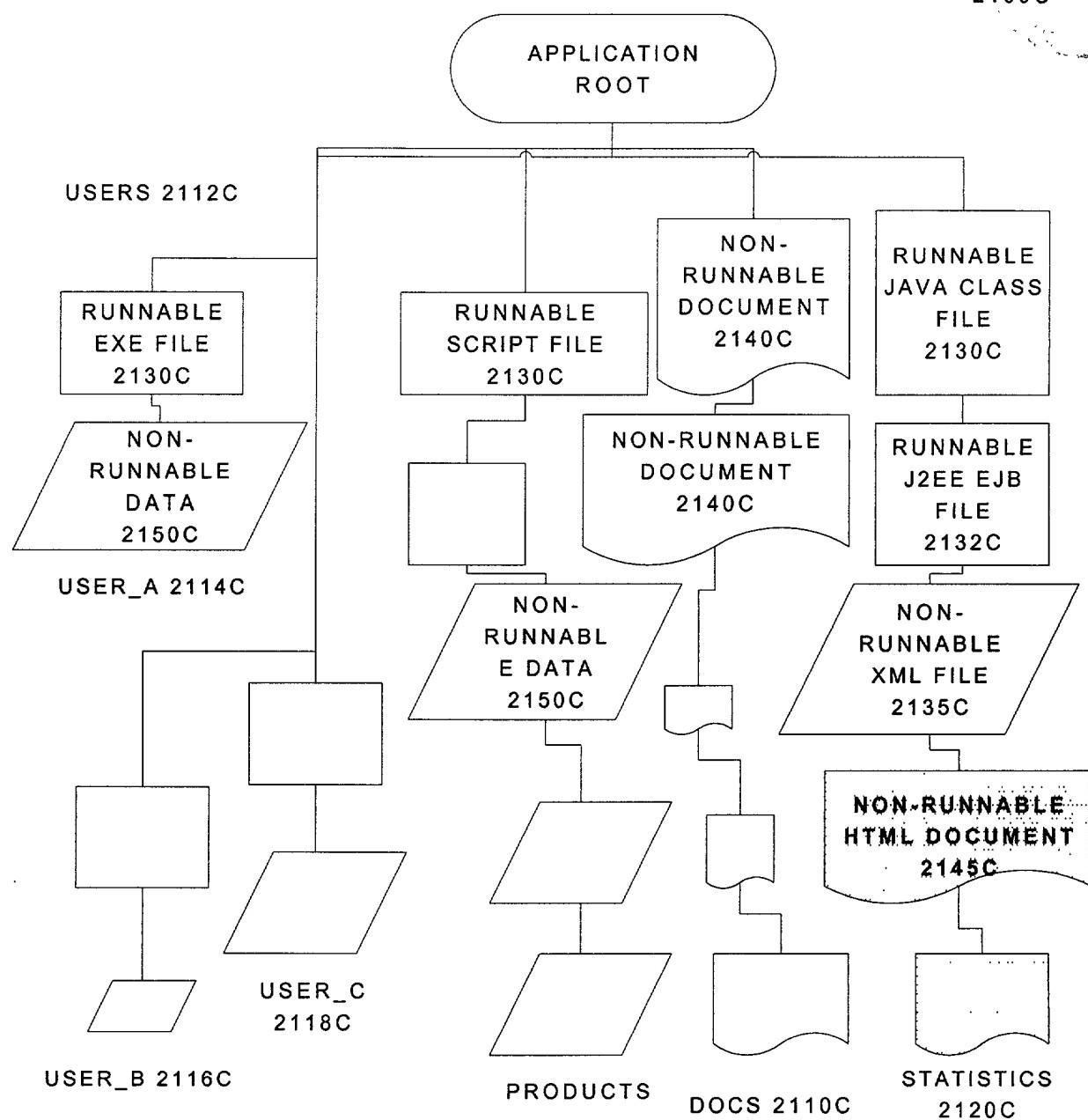
2100B

FIGURE 21B

2100C



PRIOR ART

FIGURE 21C

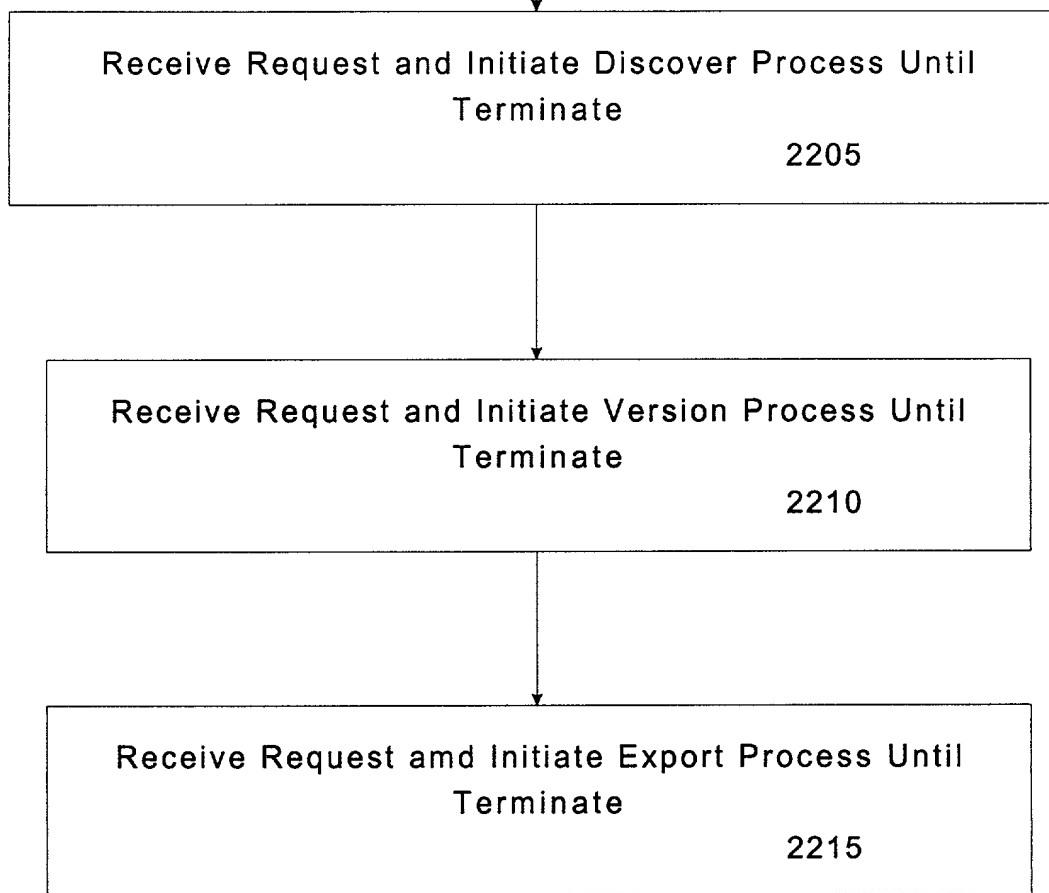
2100D

EE 220	ASSET INTERFACE 230 (OPTIONAL)	LOGIC/DATA (LD) 210



FIGURE 21D

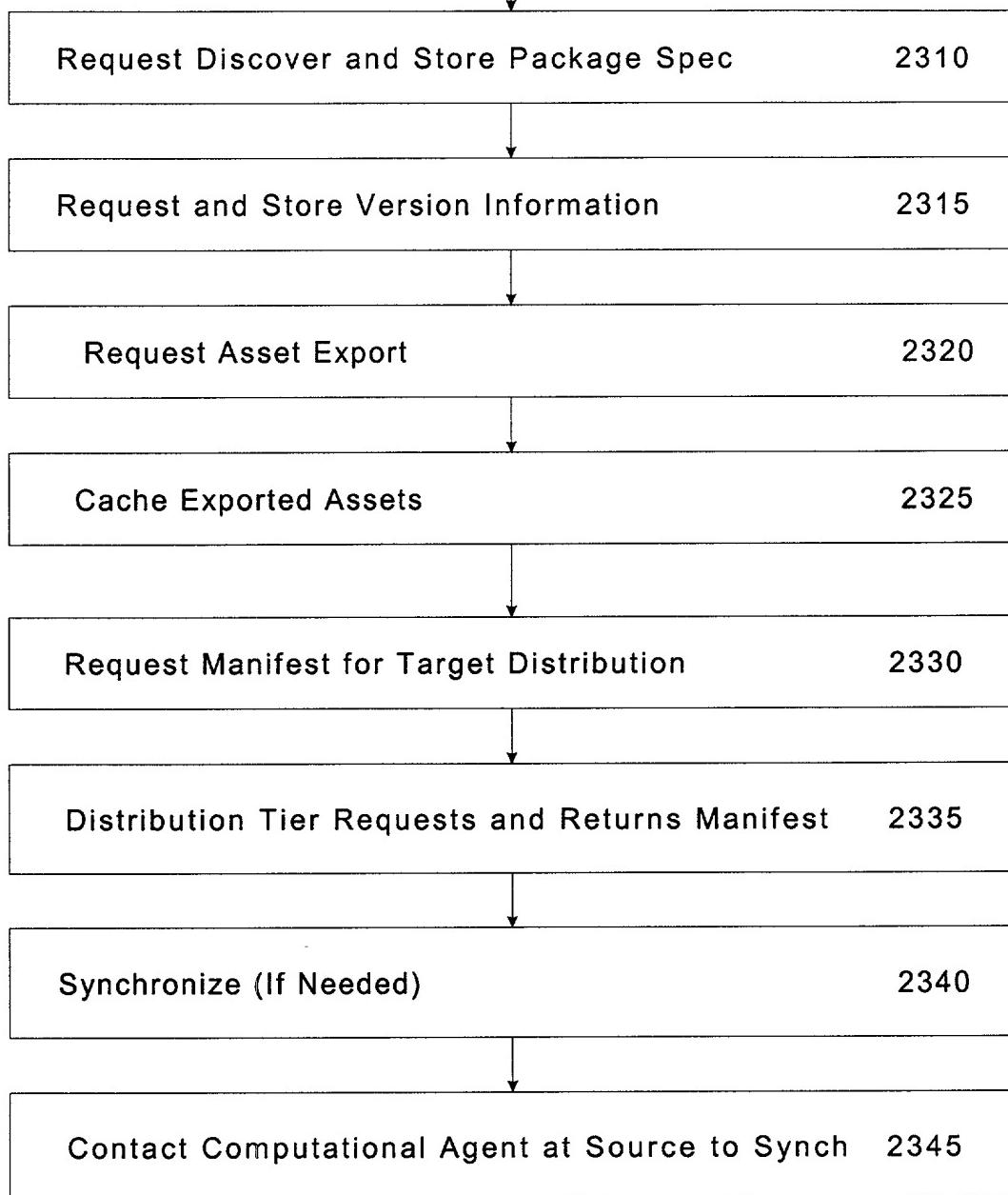
2200



Publishing Agent Method

Figure 22

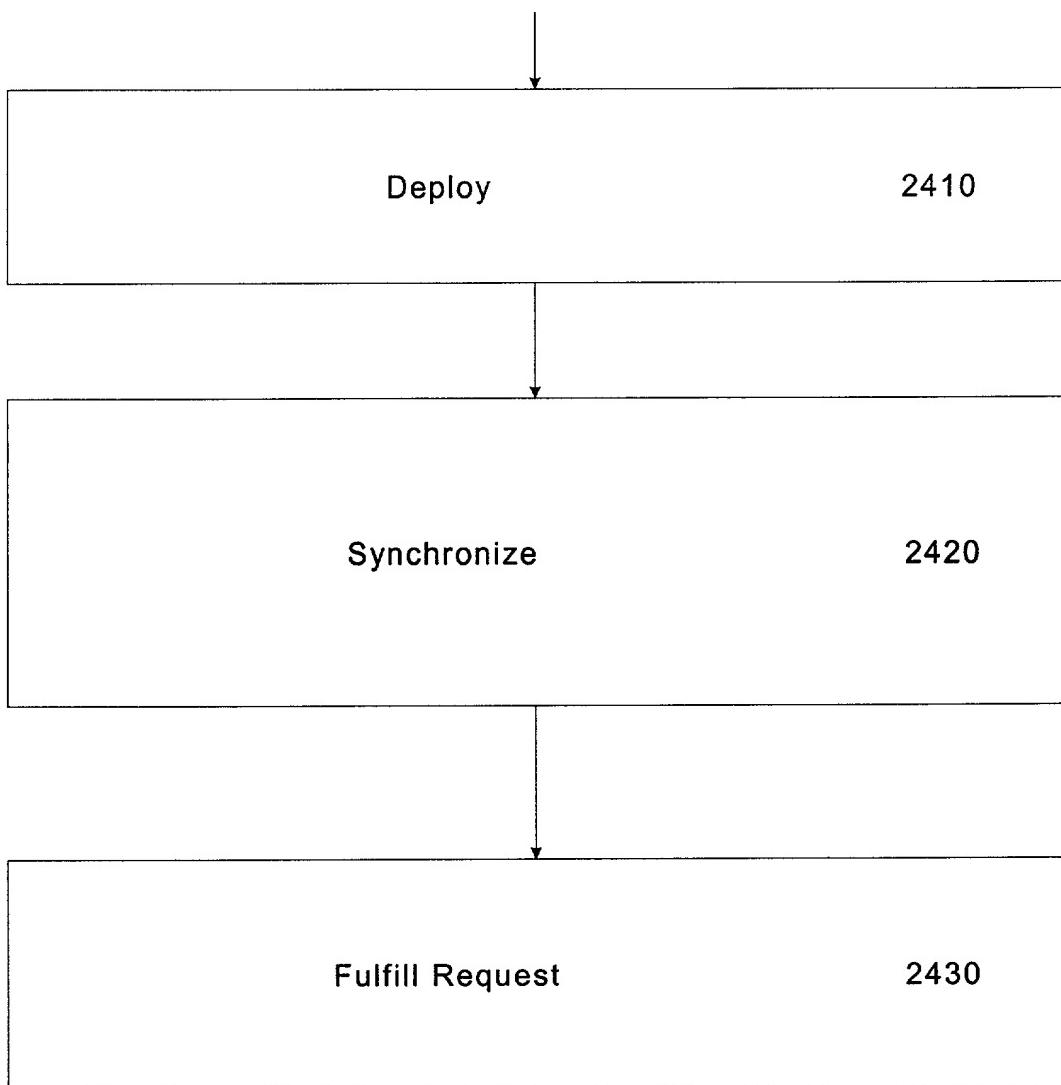
2300



### Subscriber Agent Method

Figure 23

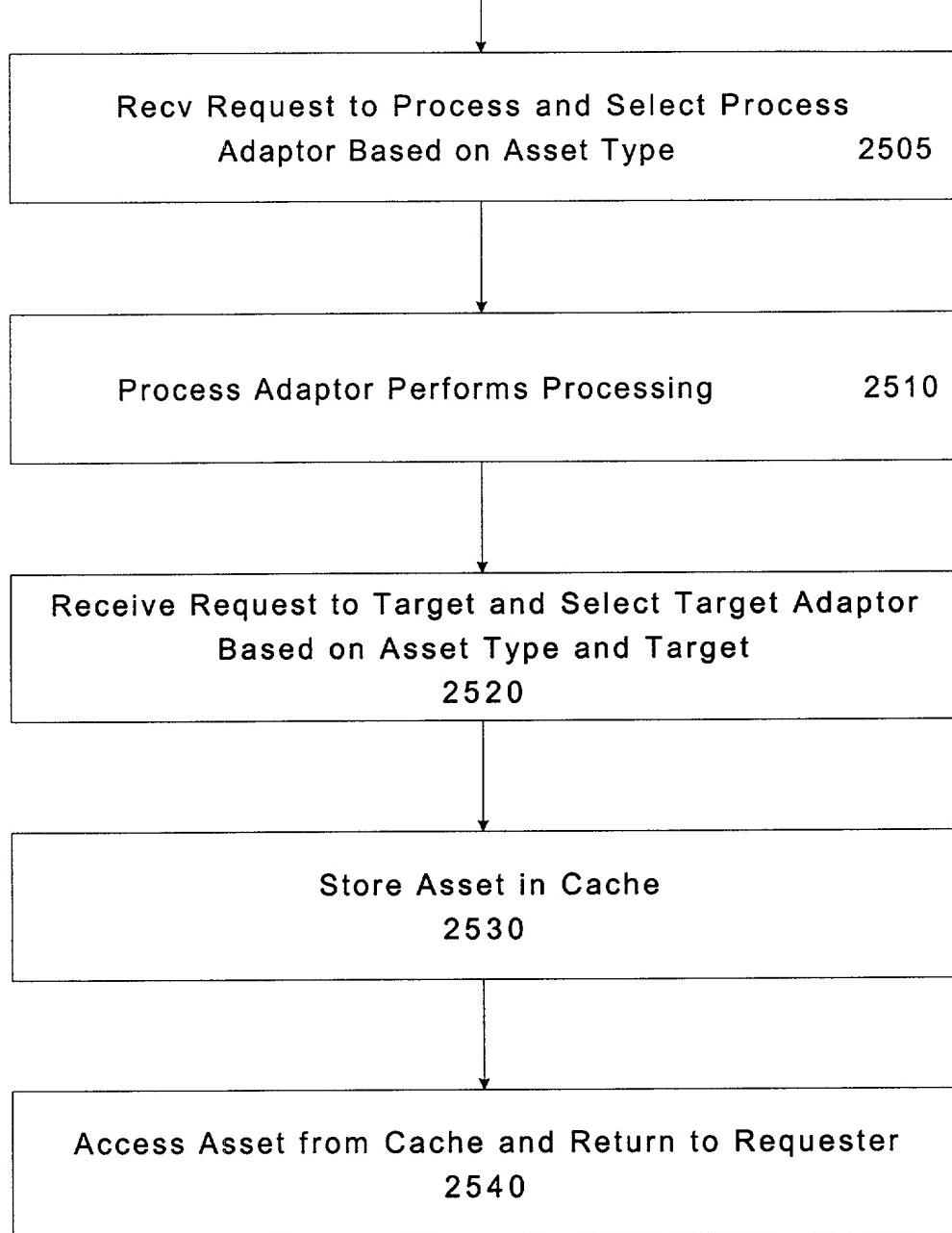
2400



Computational Agent Method

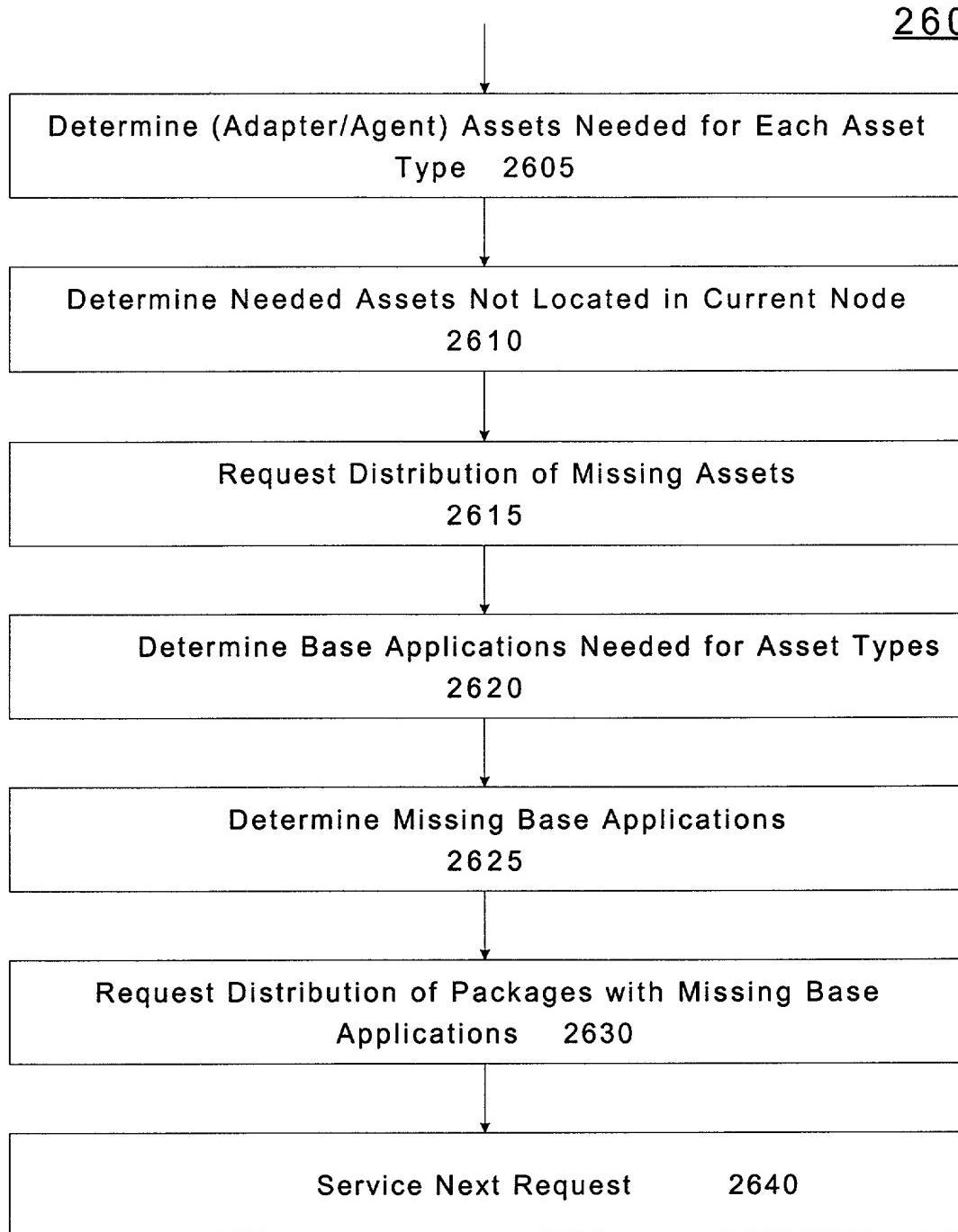
Figure 24

2500



Caching Agent Method

Figure 25



System Asset Distribution Process

Figure 26

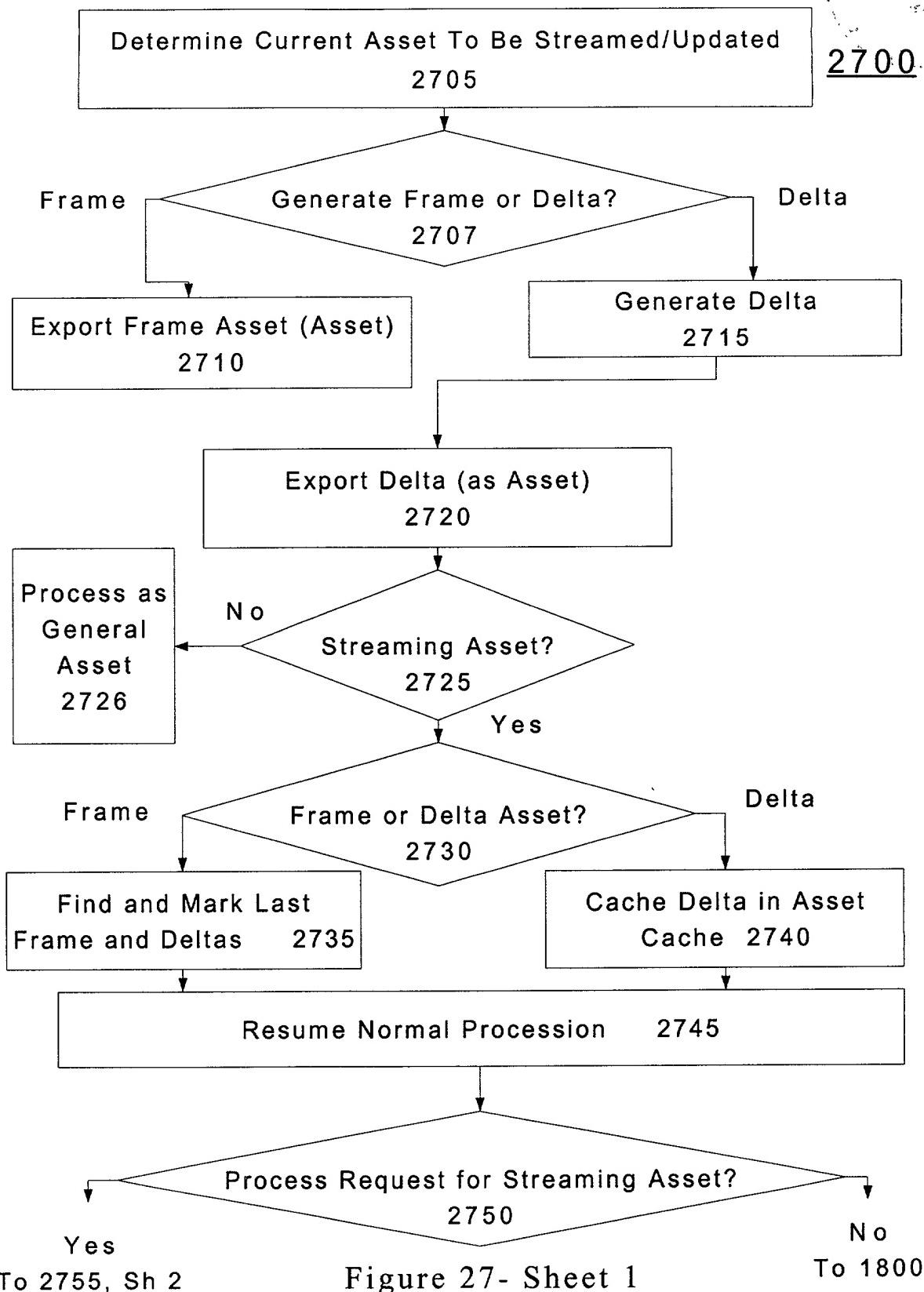


Figure 27- Sheet 1

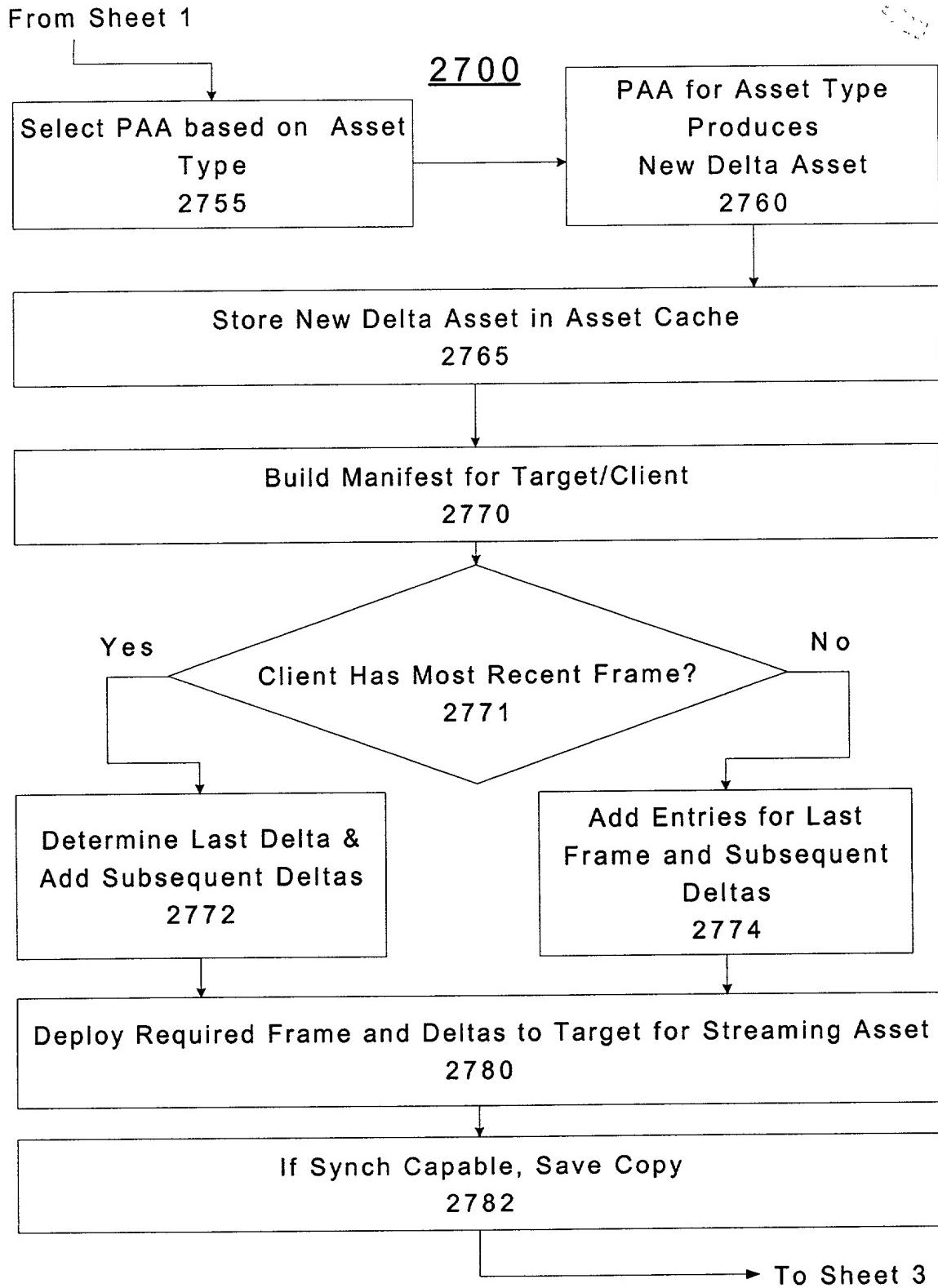


Figure 27-Sheet 2

2700

From Sheet 2

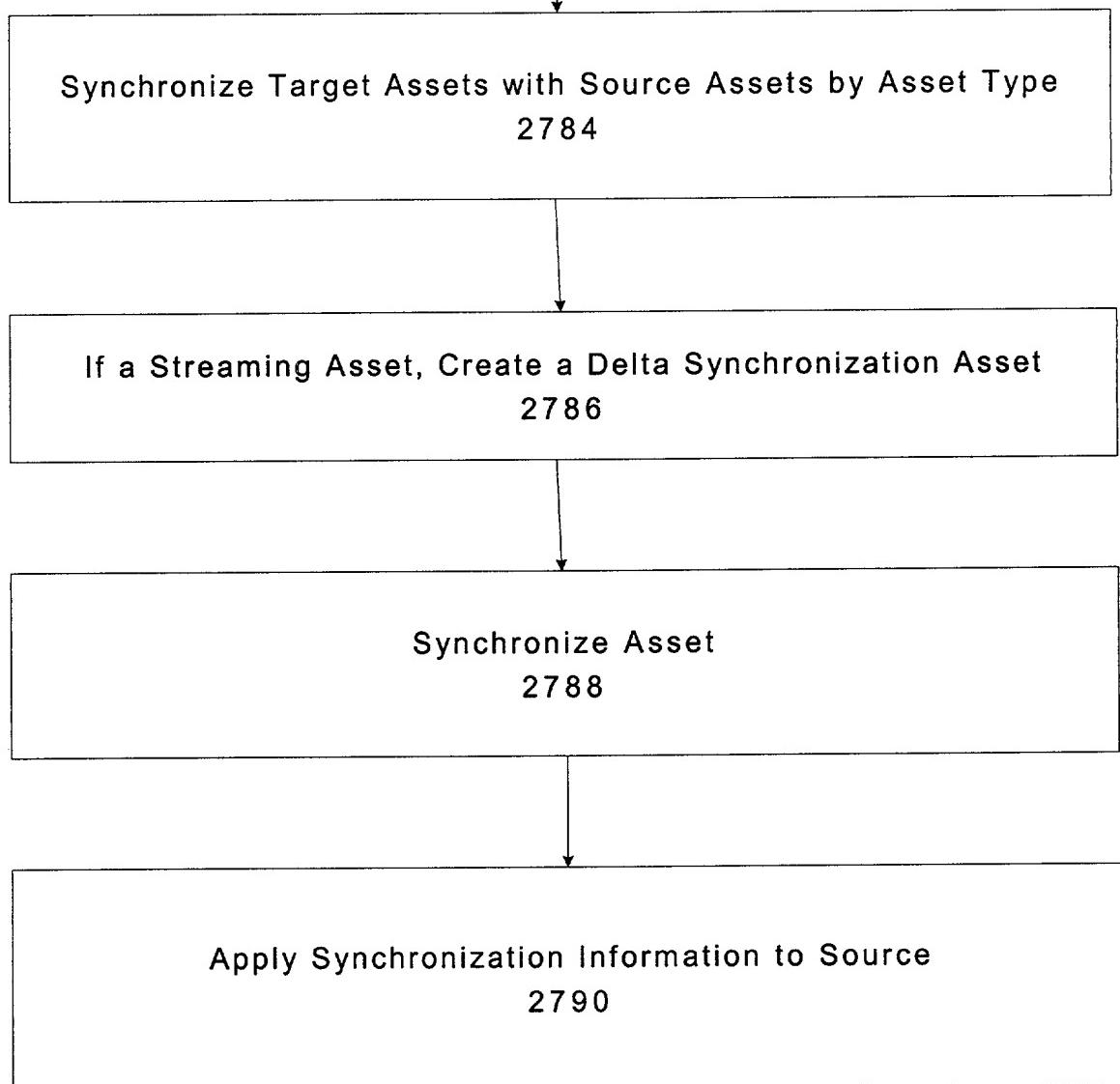
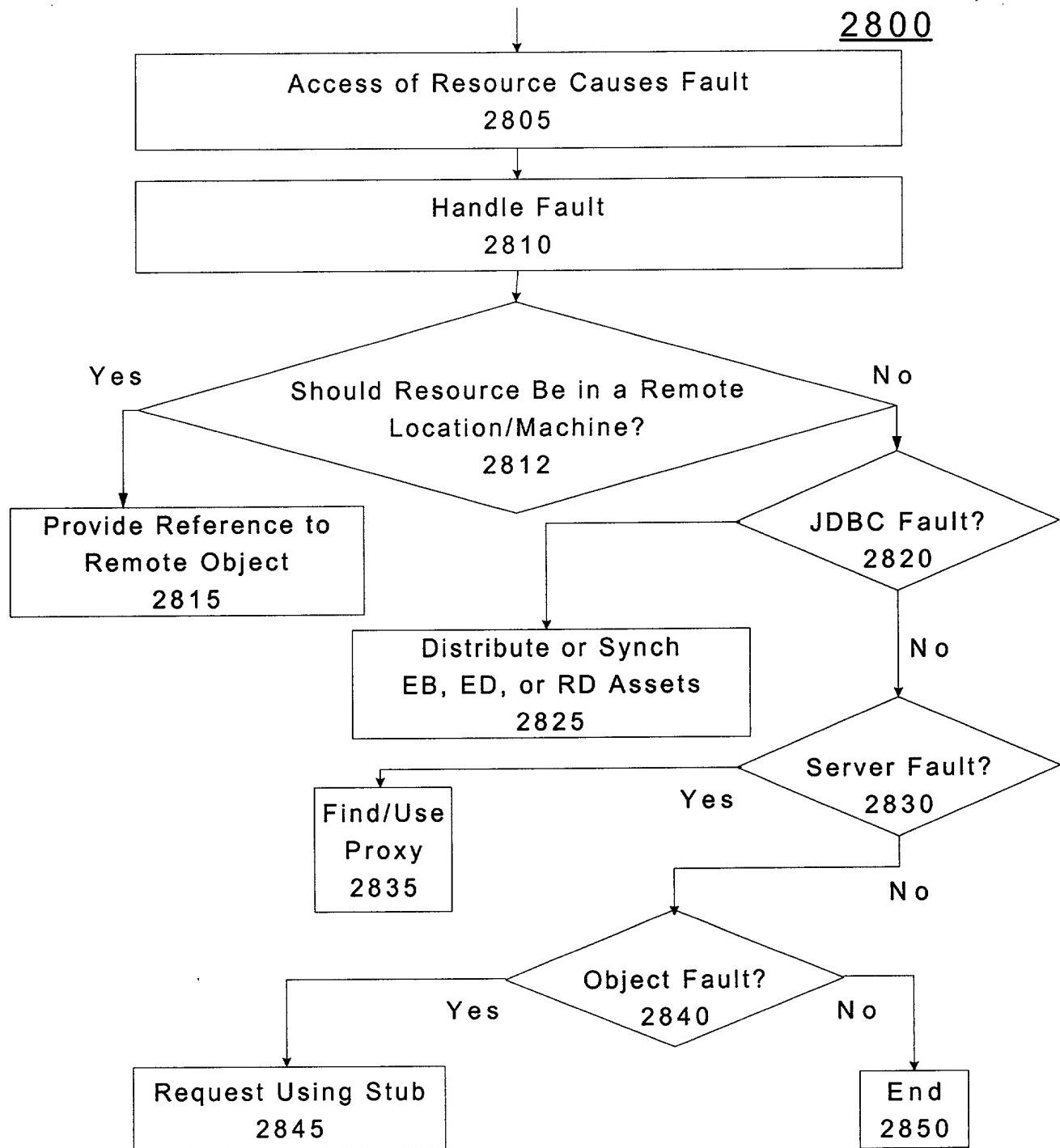


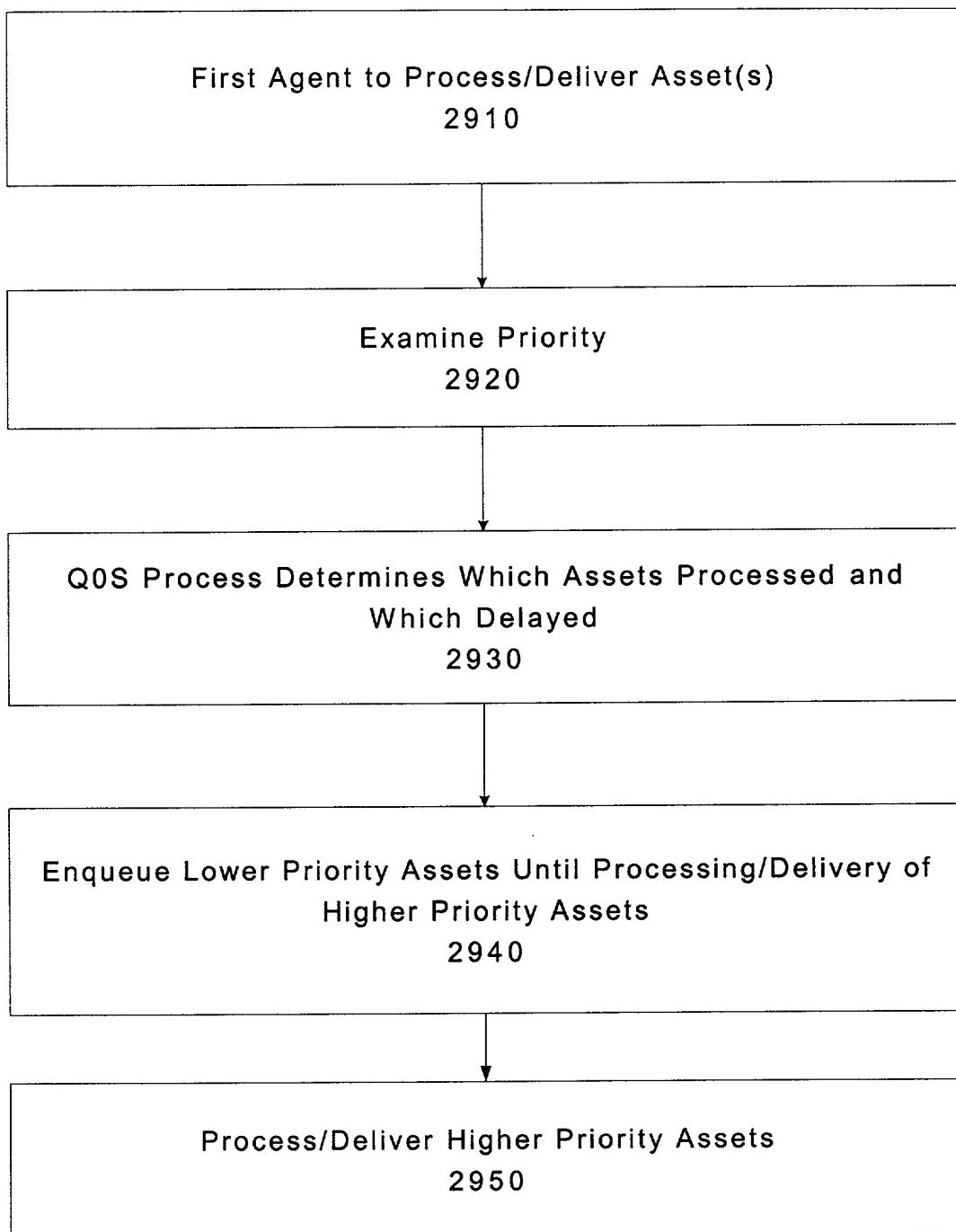
Figure 27-Sheet 3



Bridging Process

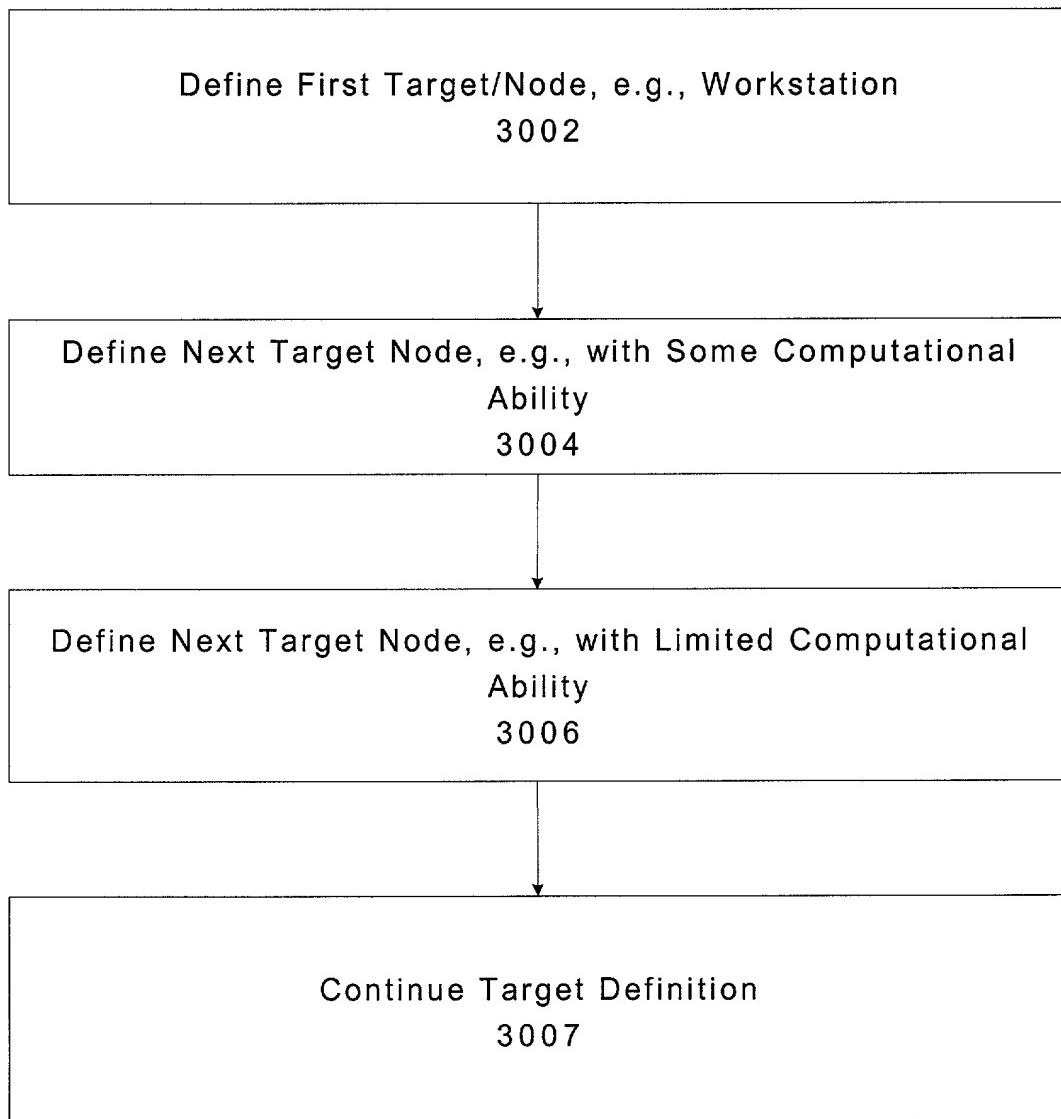
Figure 28

2900



QoS  
Figure 29

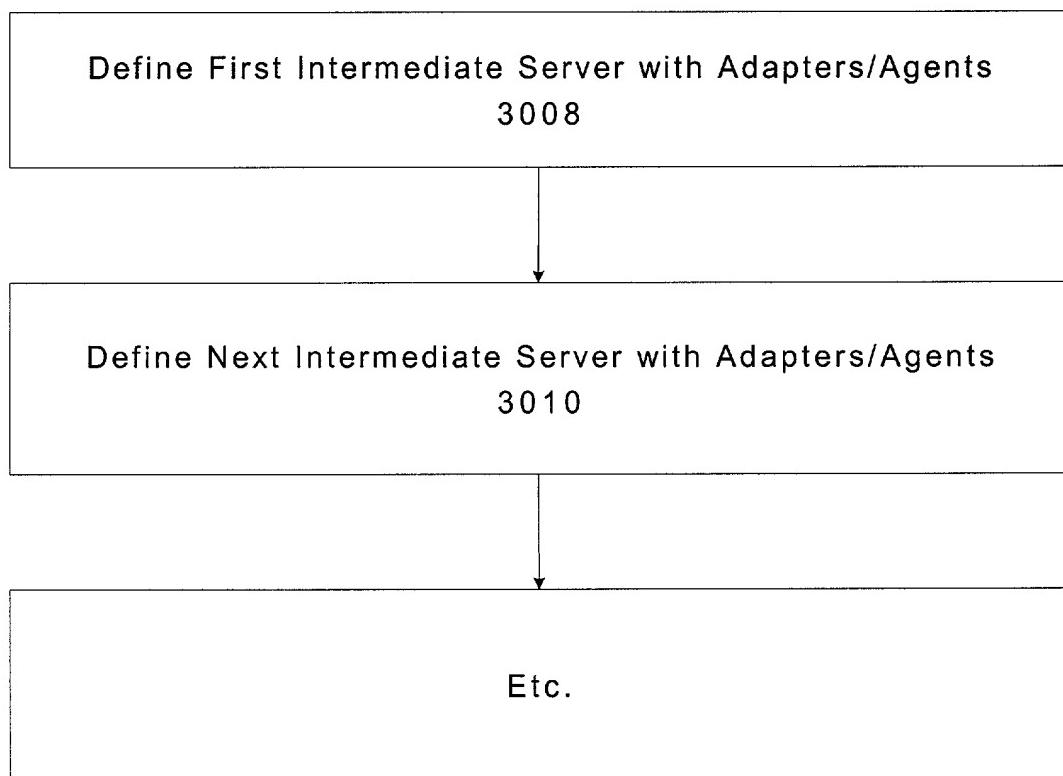
3000A



Target/Client Definition

Figure 30A

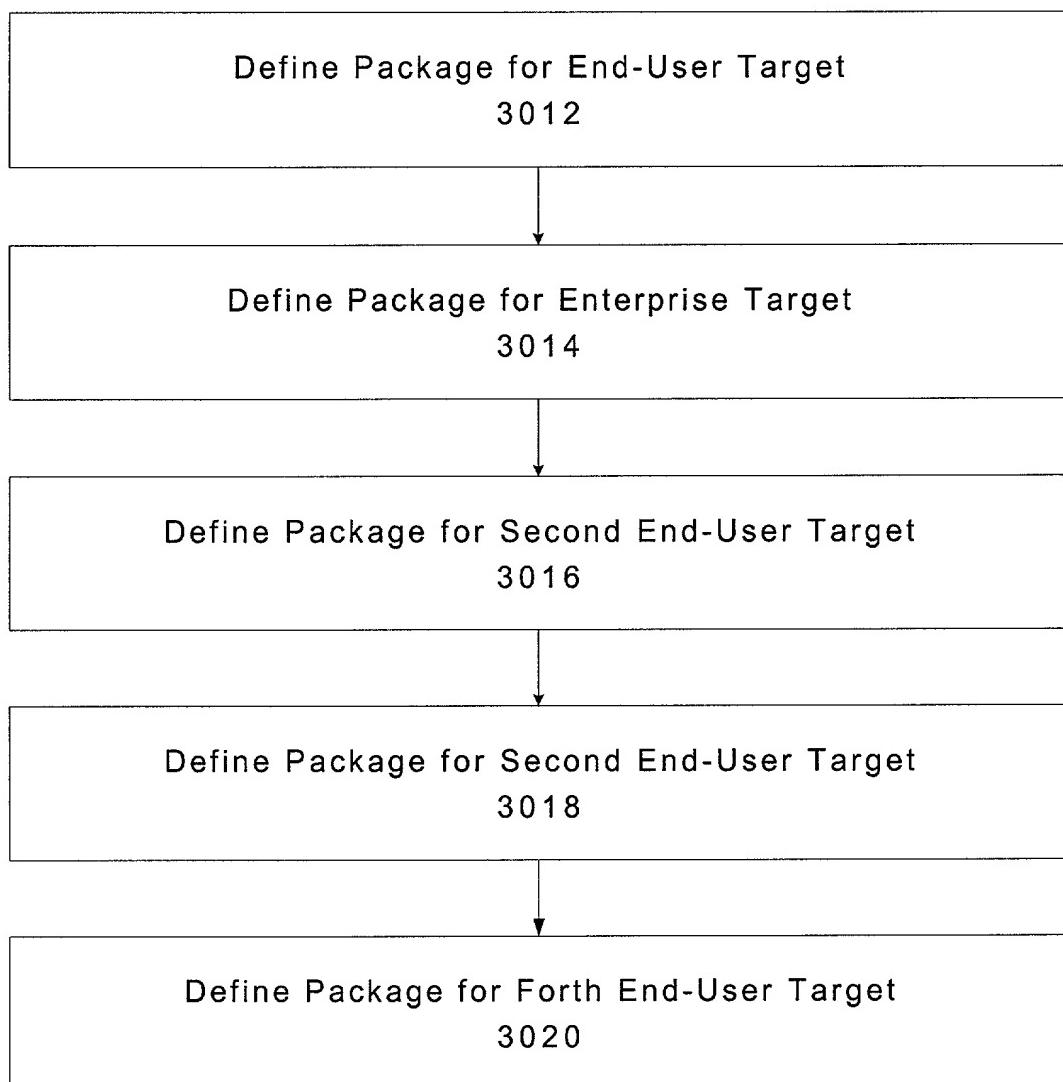
3000B



Server Definition

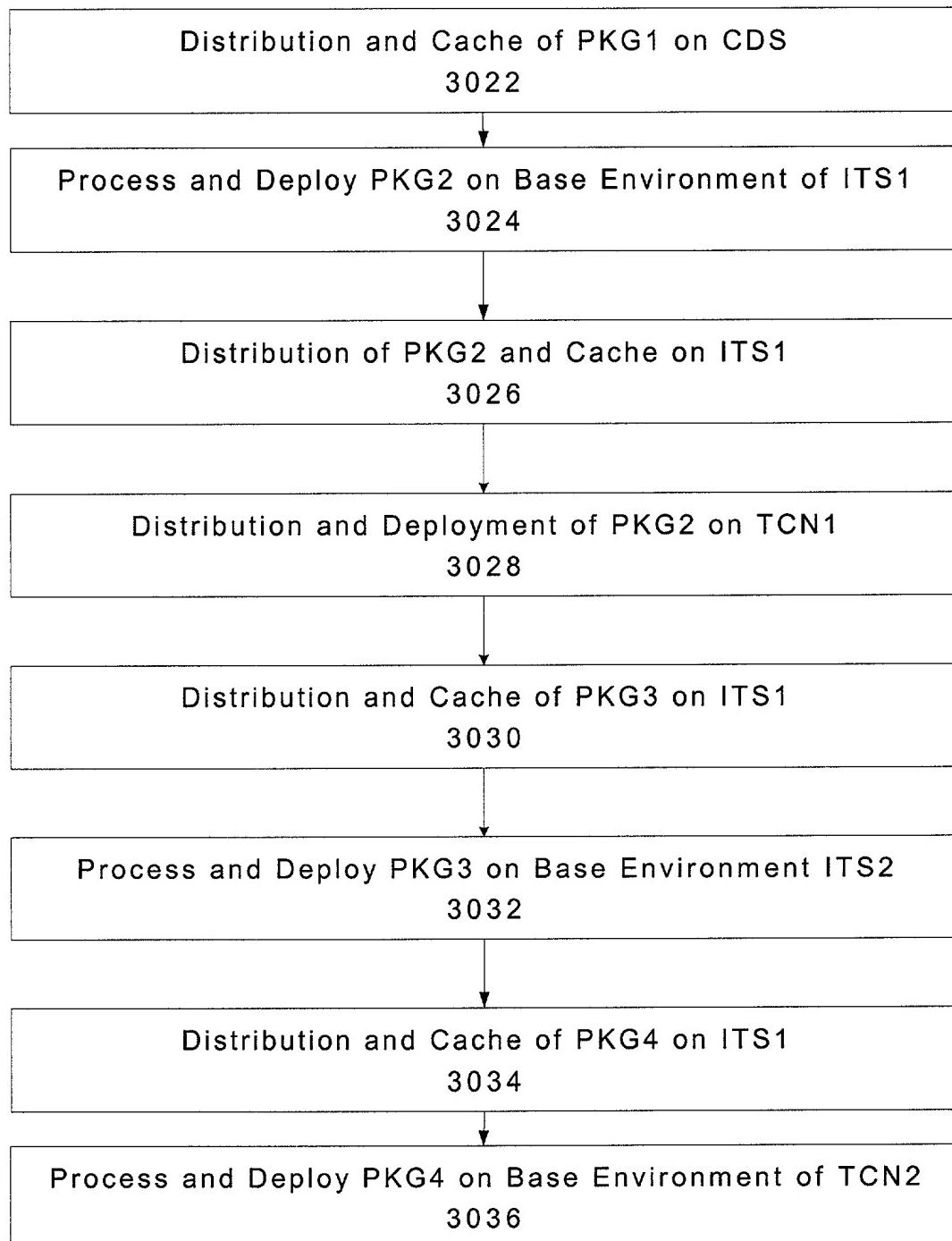
Figure 30B

3000C



Define Packages/Applications

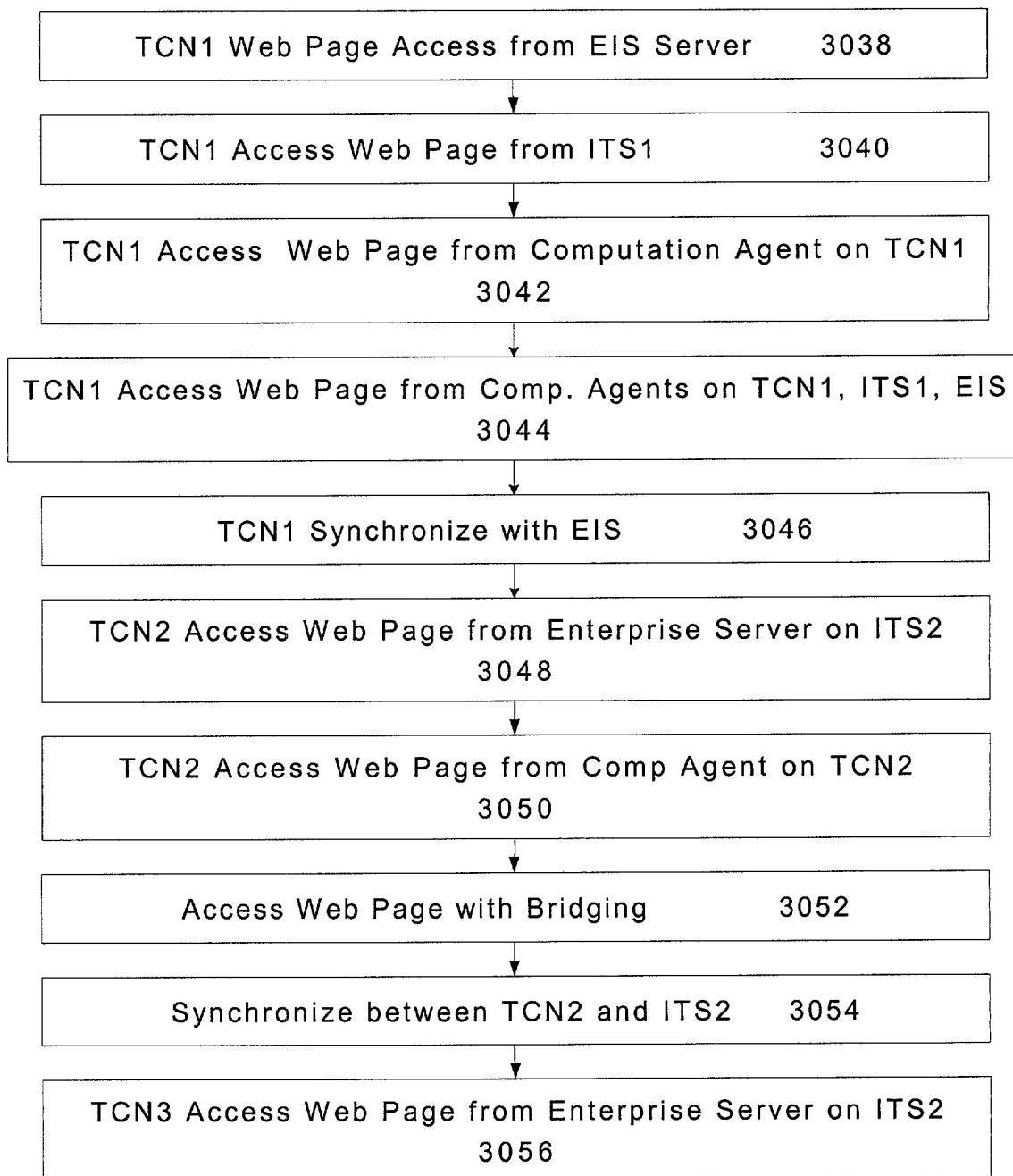
Figure 30C

3000D

Distributing to Computational Environments

Figure 30D

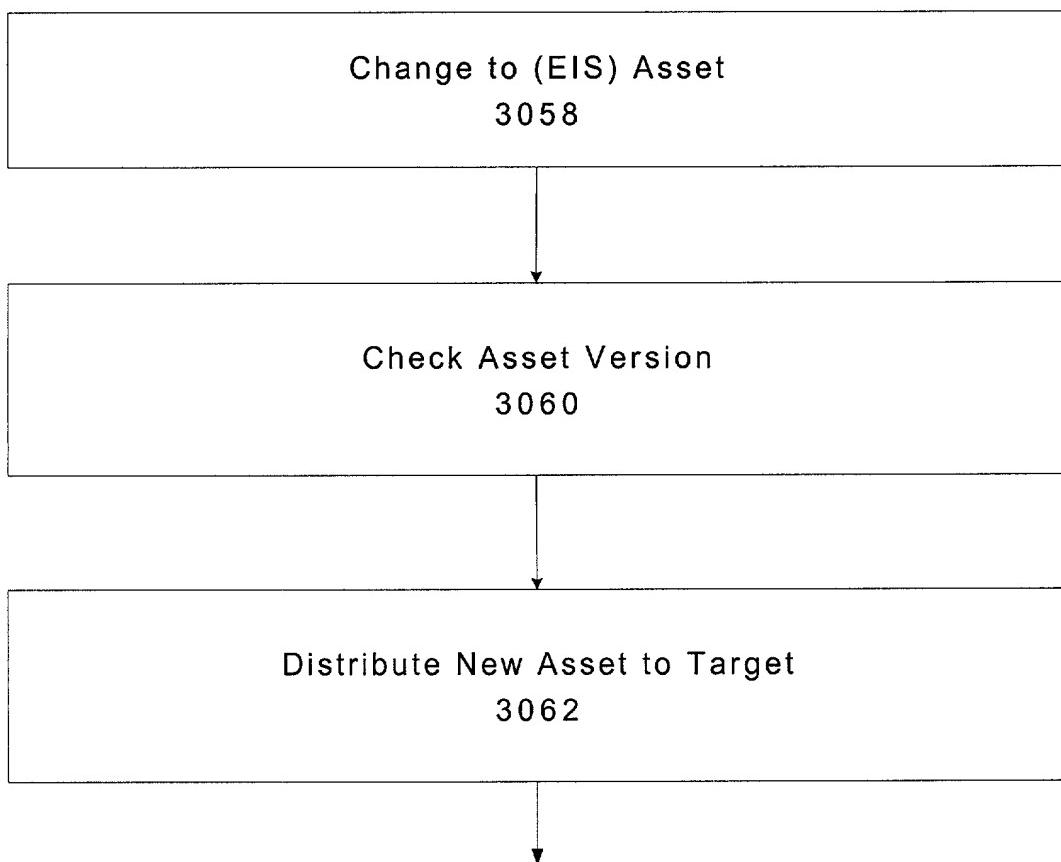
3000E



Distributed Execution of Assets

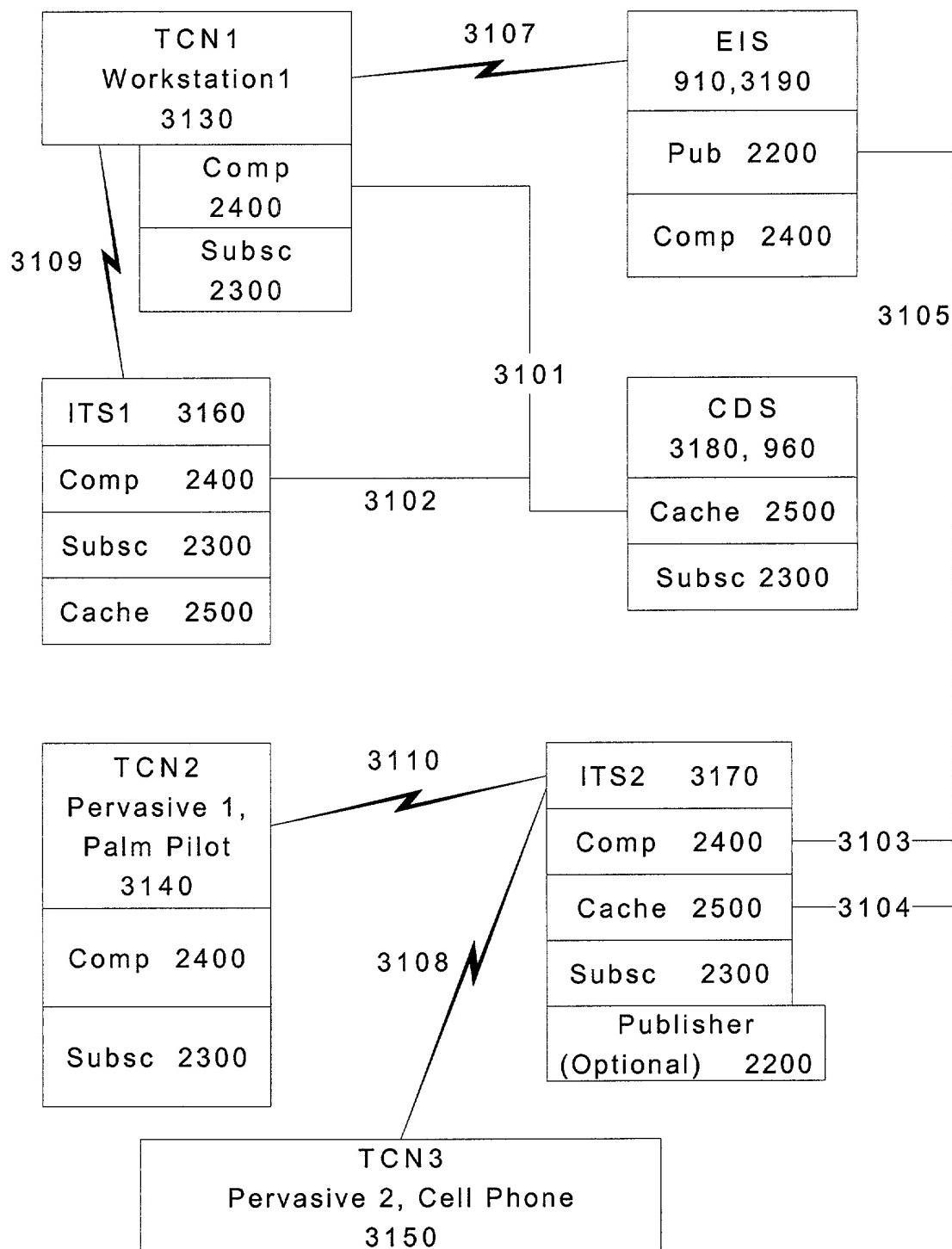
Figure 30E

3000F



Distribution of Current Assets

Figure 30F

3100

Example Network Connections and Asset Distribution

Figure 31

3200

BI-DIRECTIONAL FLOW 3203

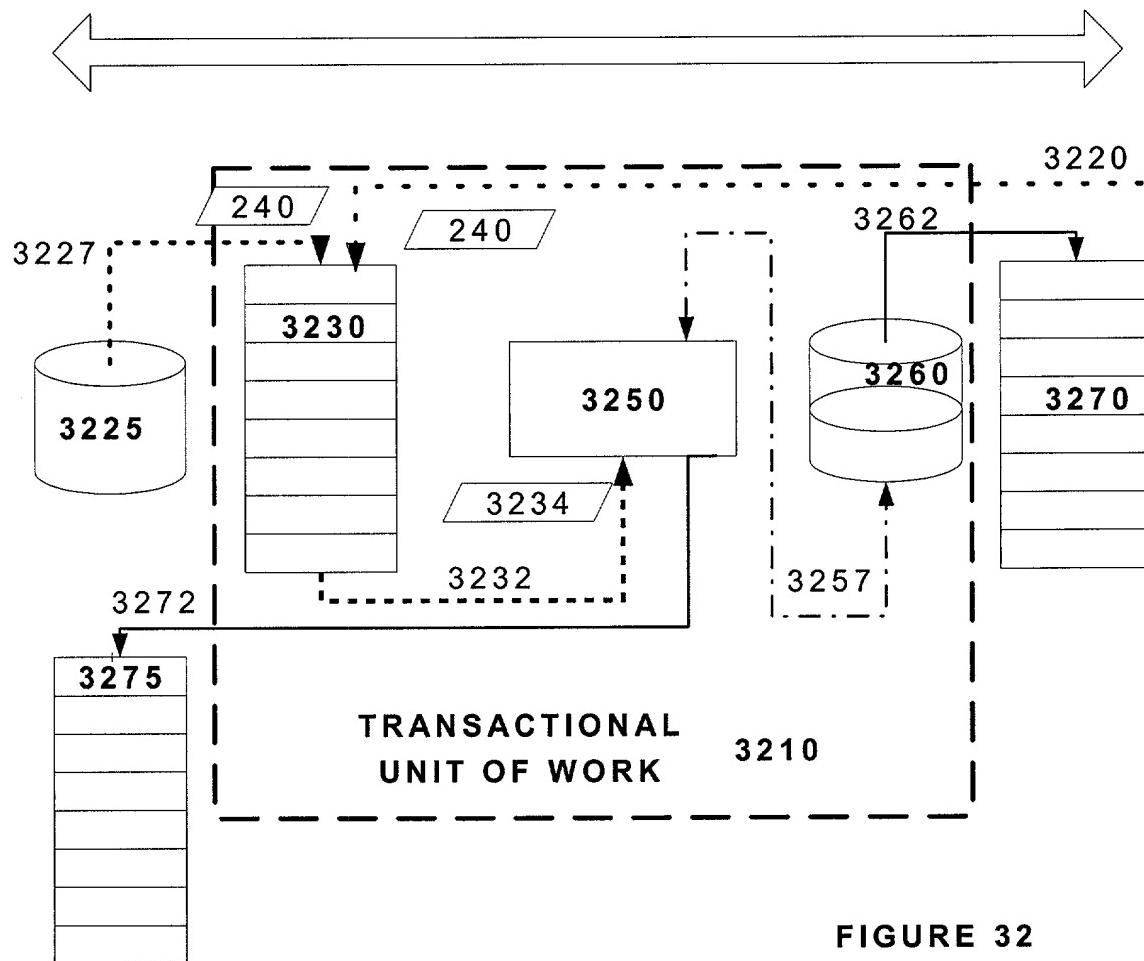
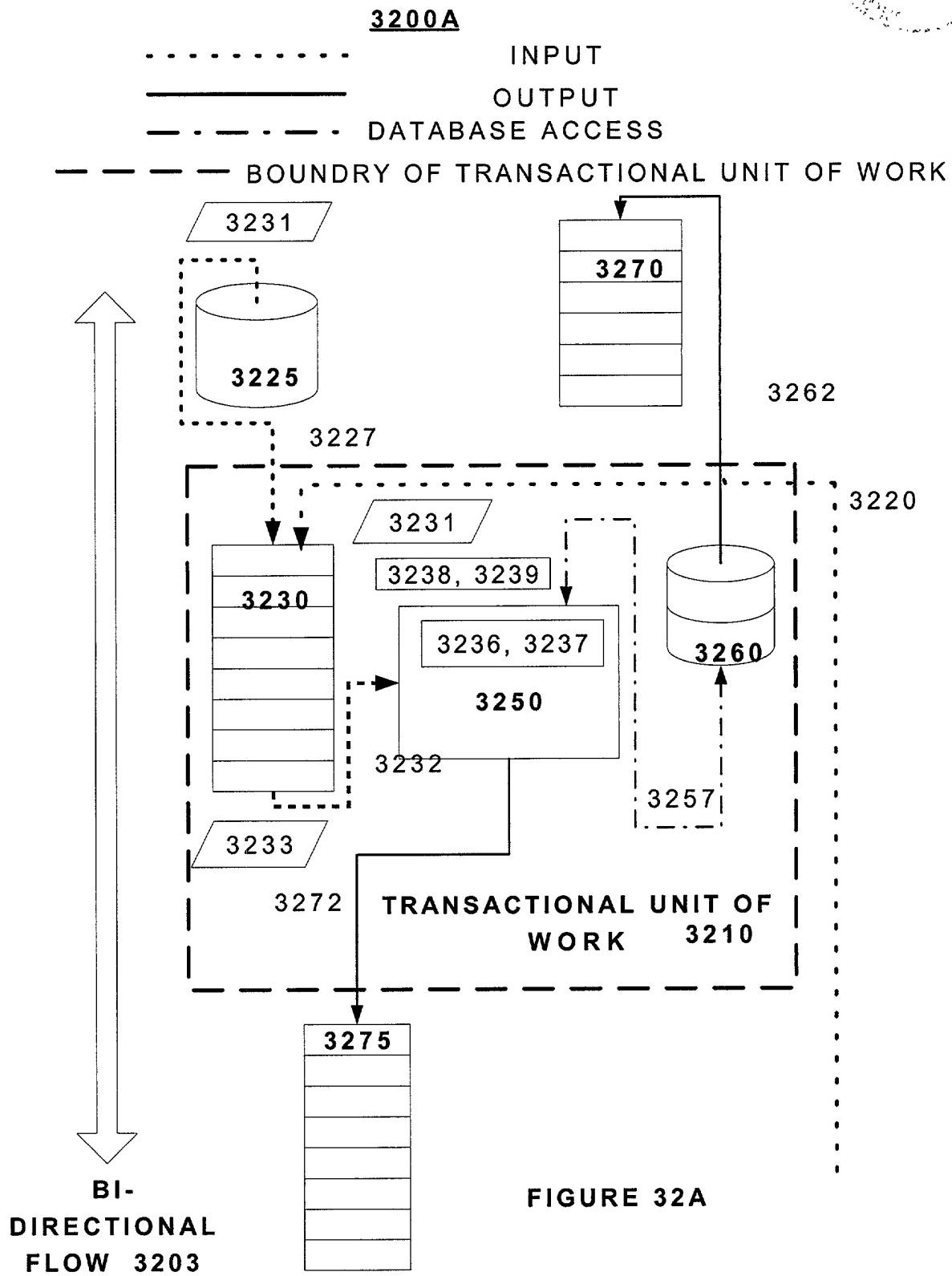
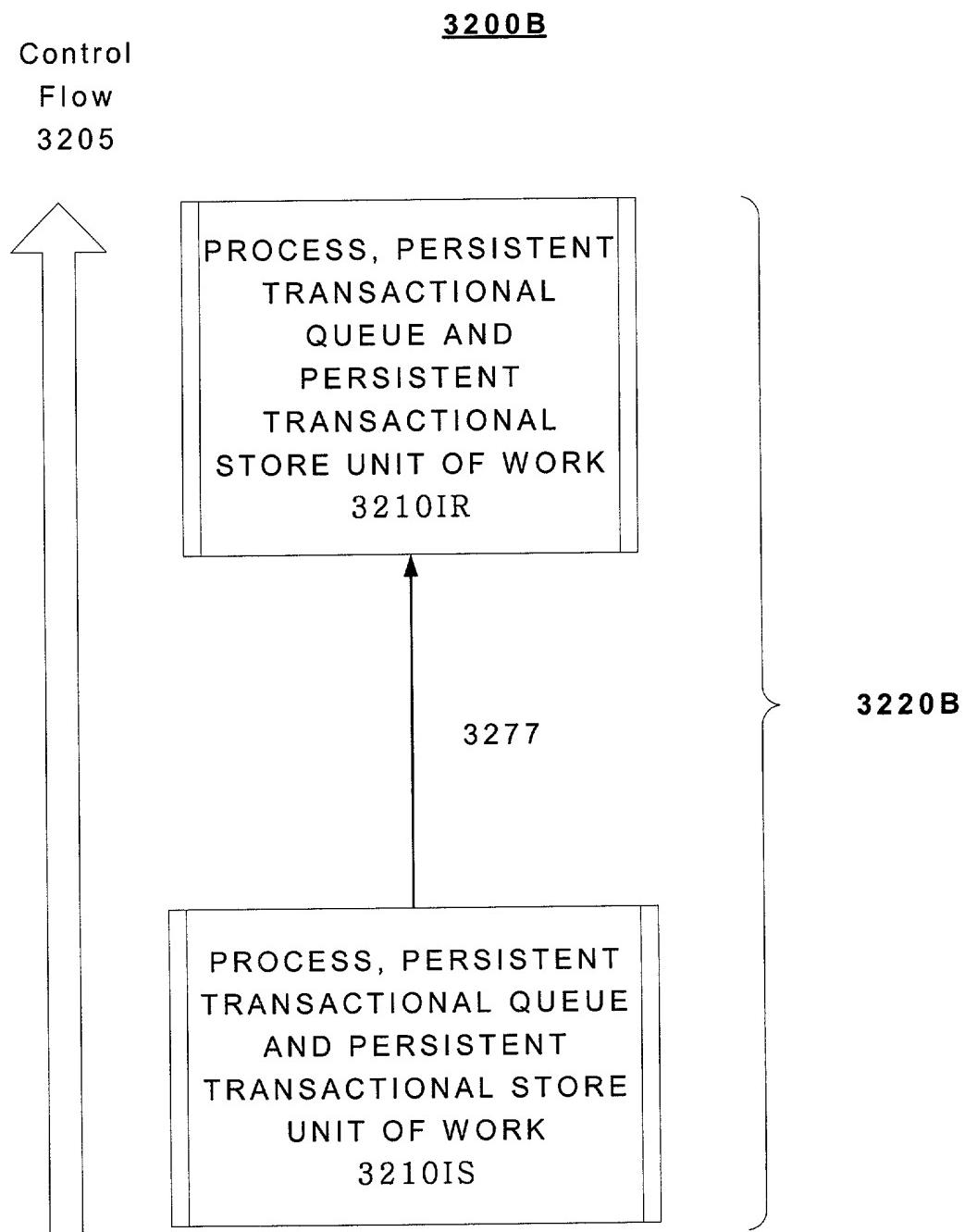


FIGURE 32

**FIGURE 32A**



**FIGURE 32B**

DETAILED DESCRIPTION

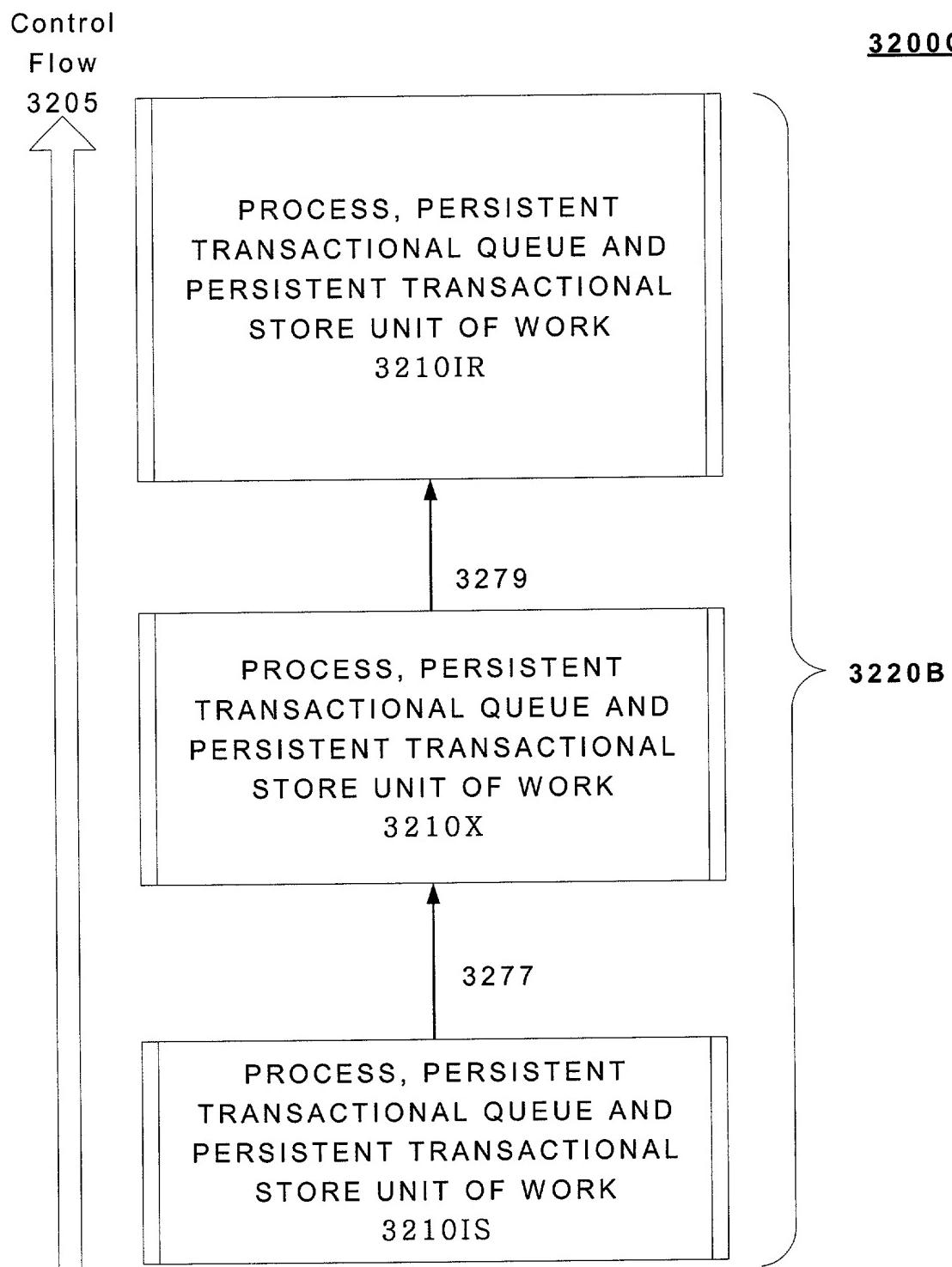


FIGURE 32C

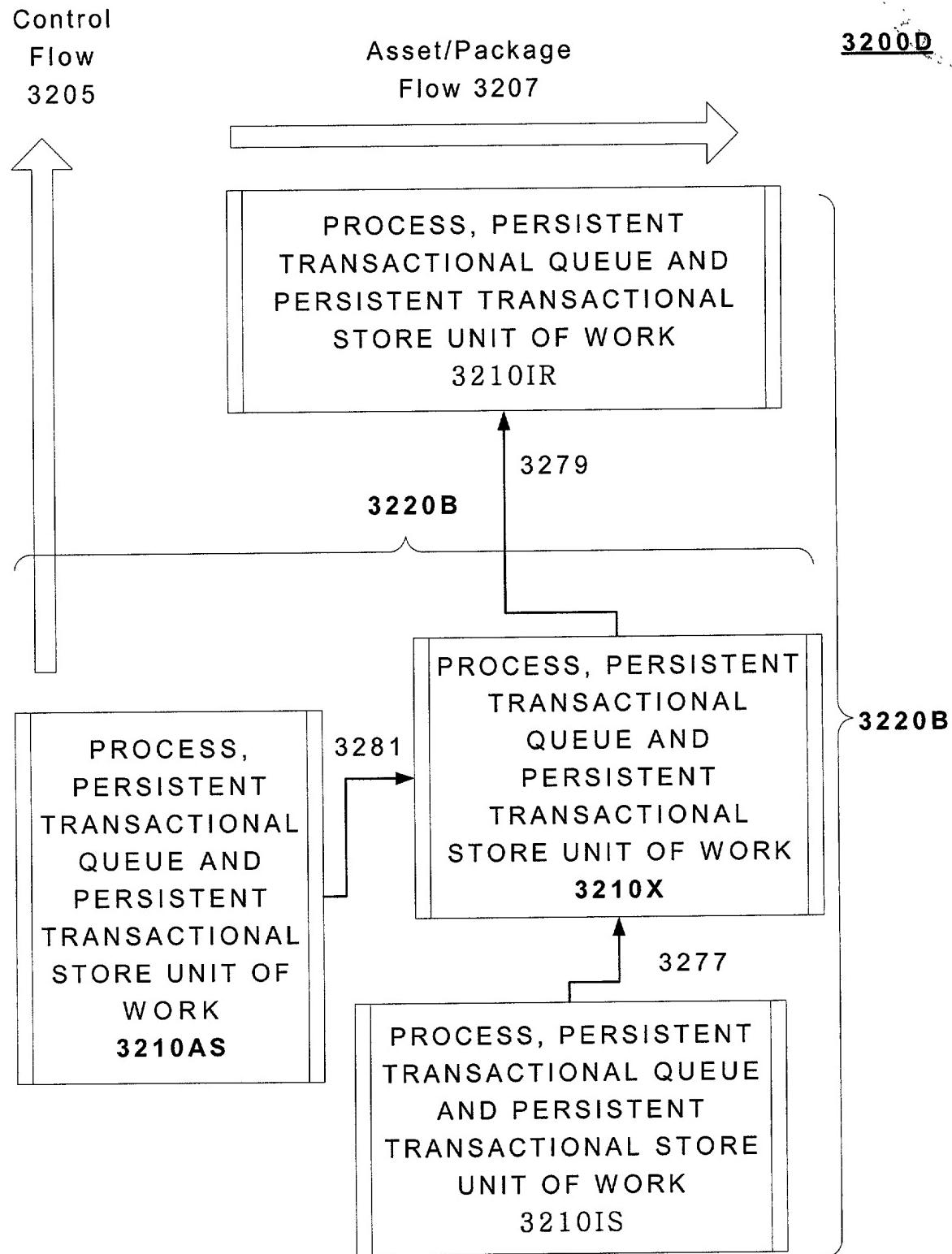


FIGURE 32D

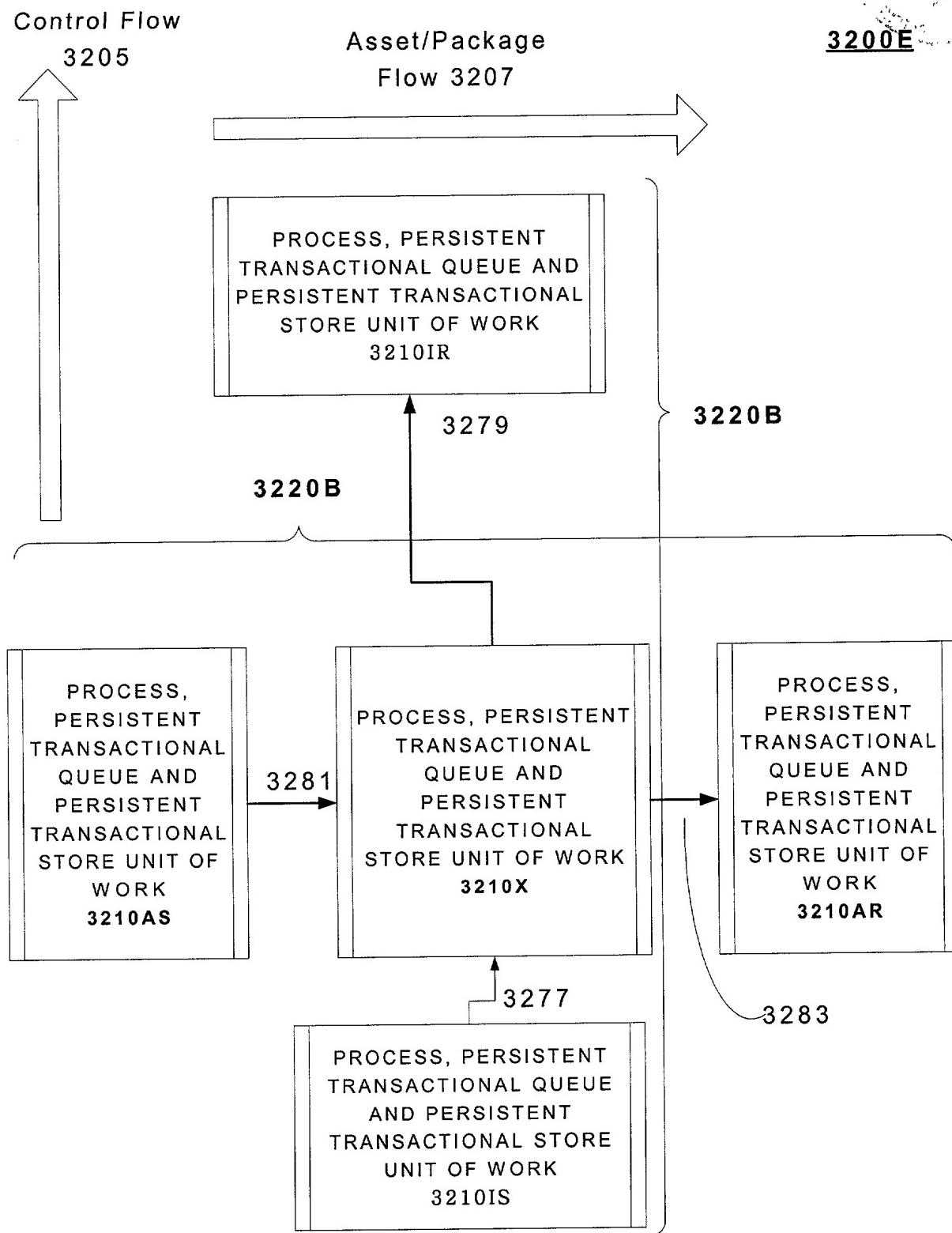


FIGURE 32E



3200F

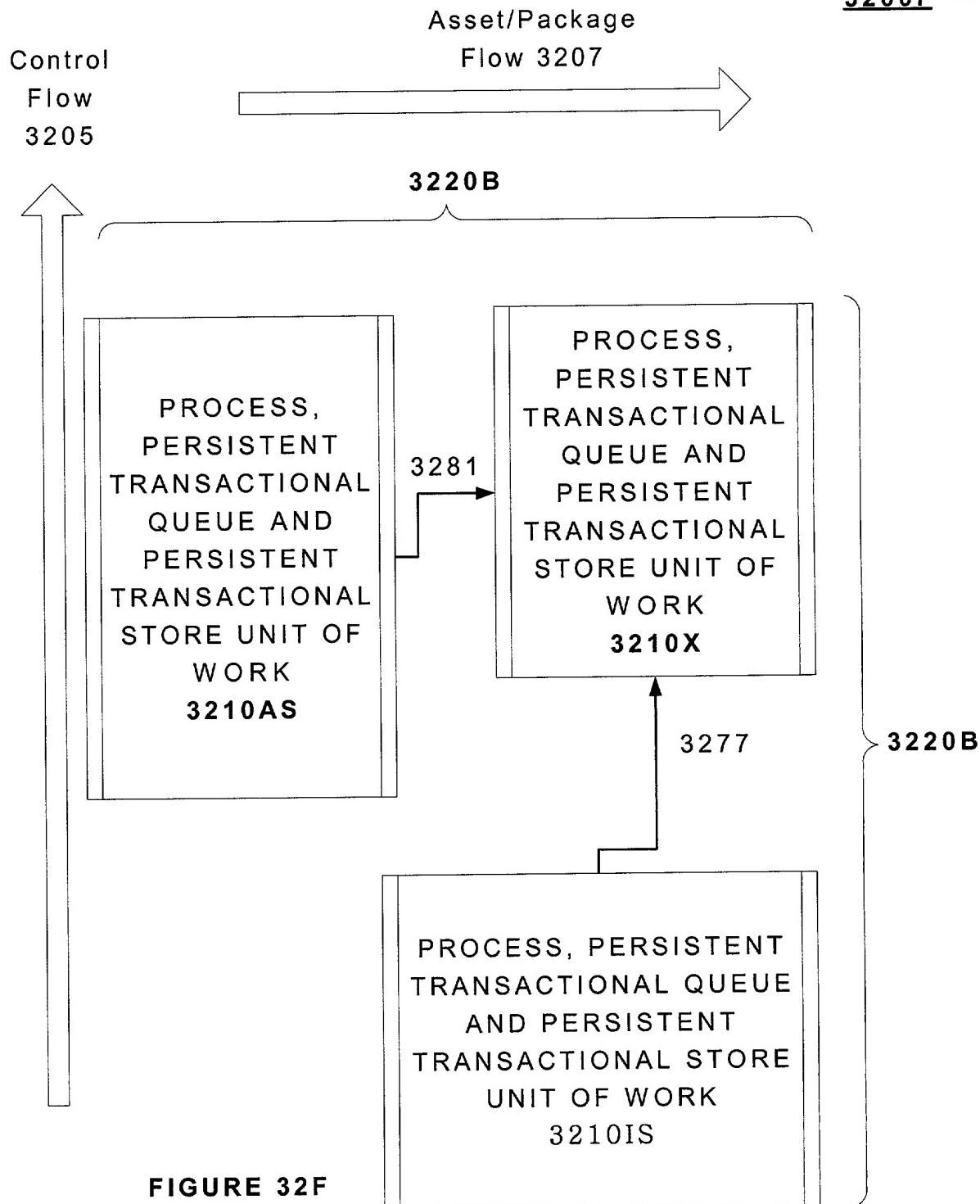
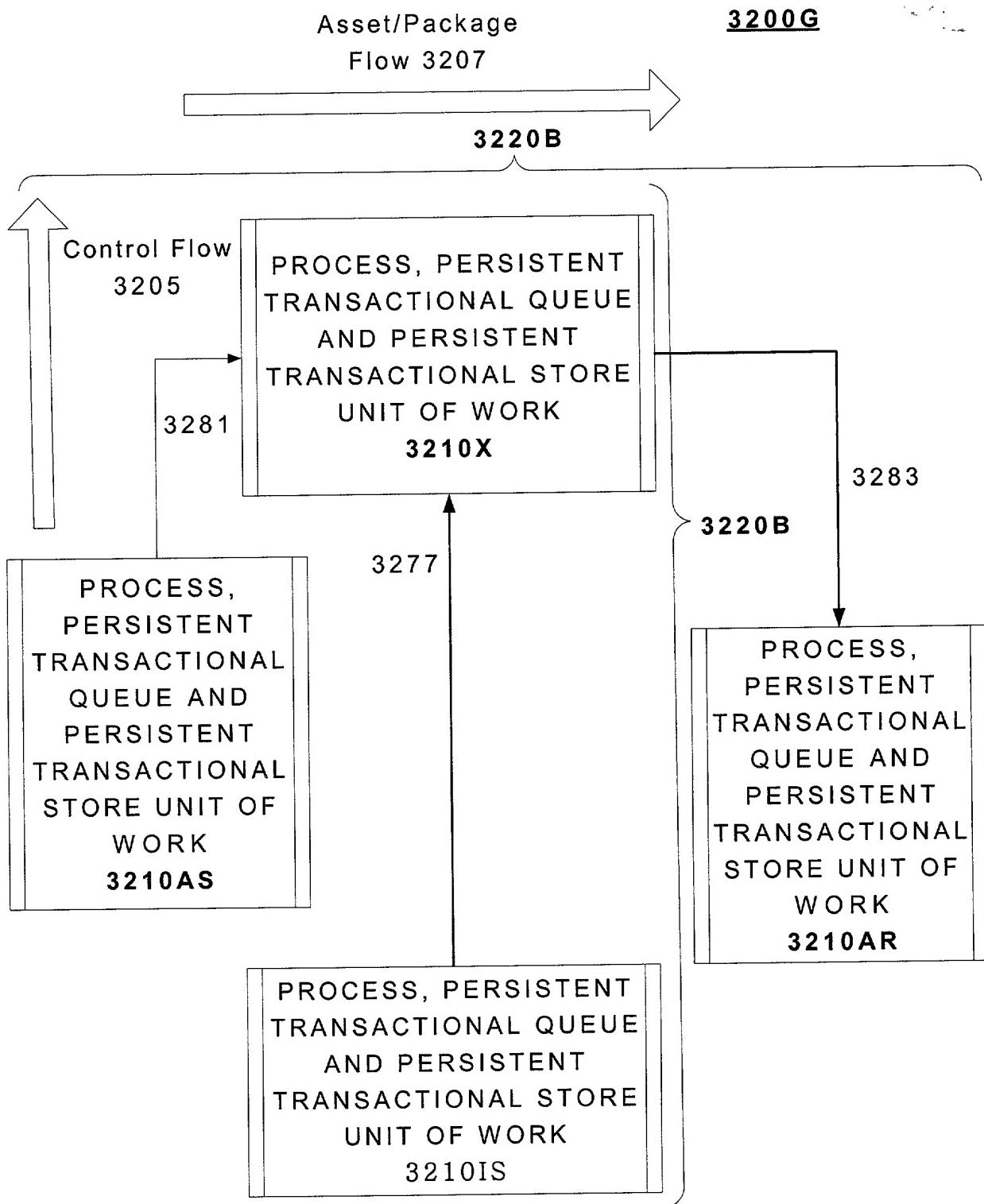


FIGURE 32F

**FIGURE 32G**

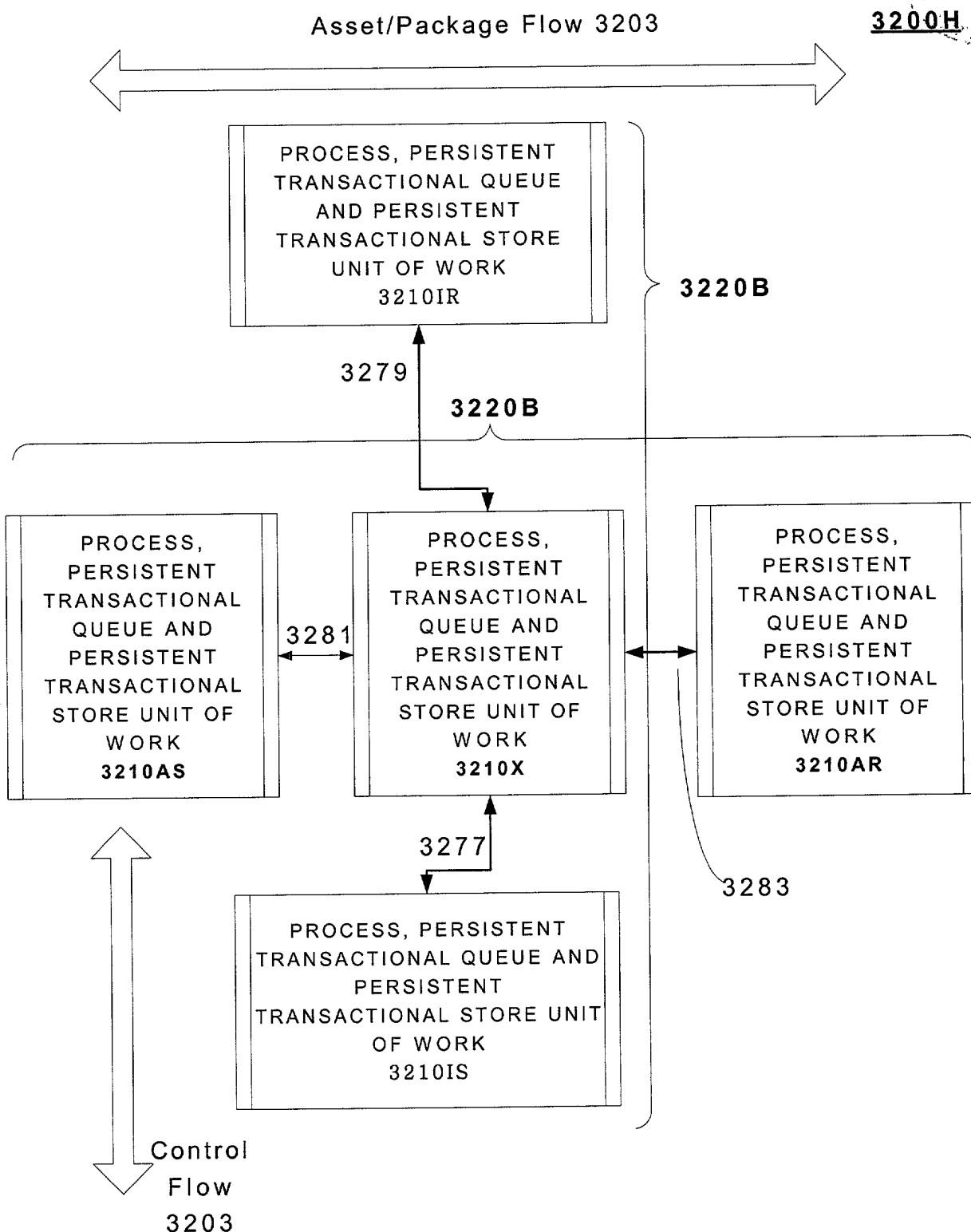


FIGURE 32H

3300

3220B

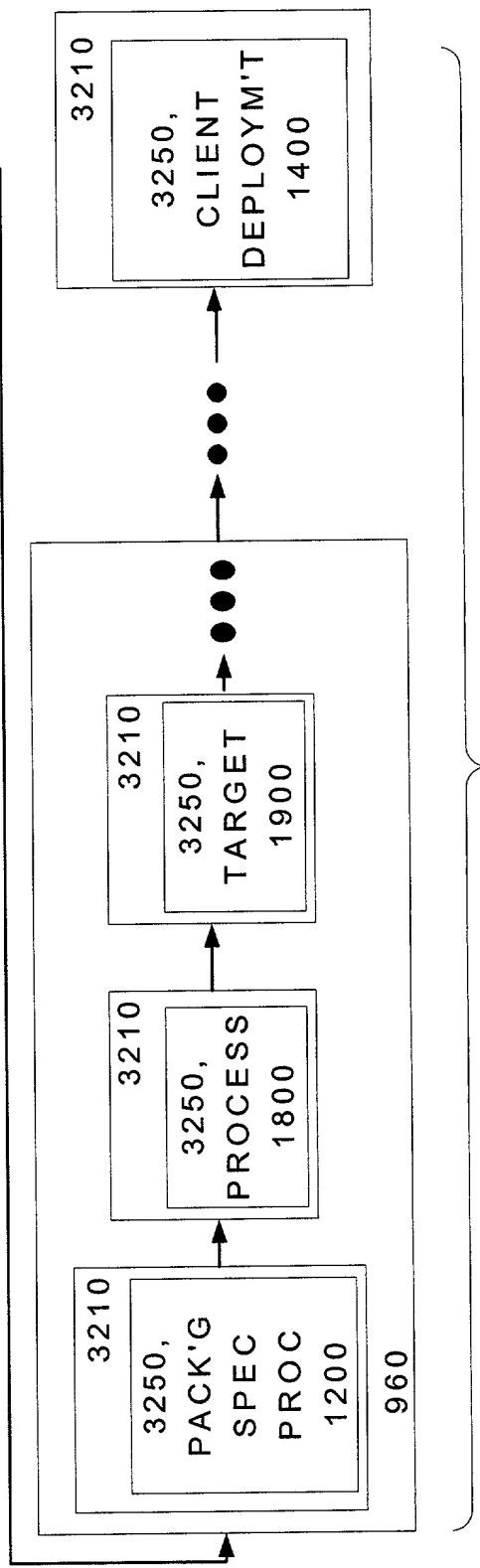
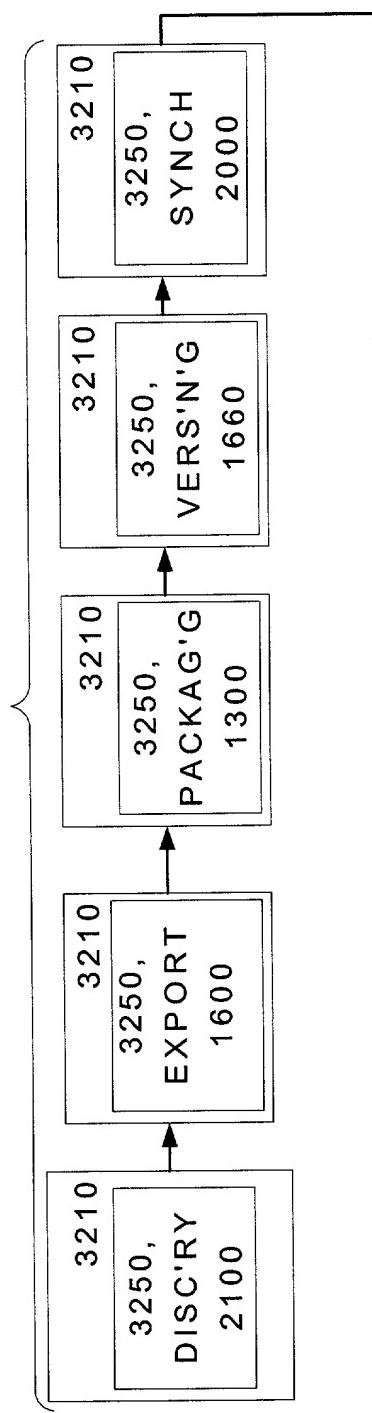


FIGURE 33

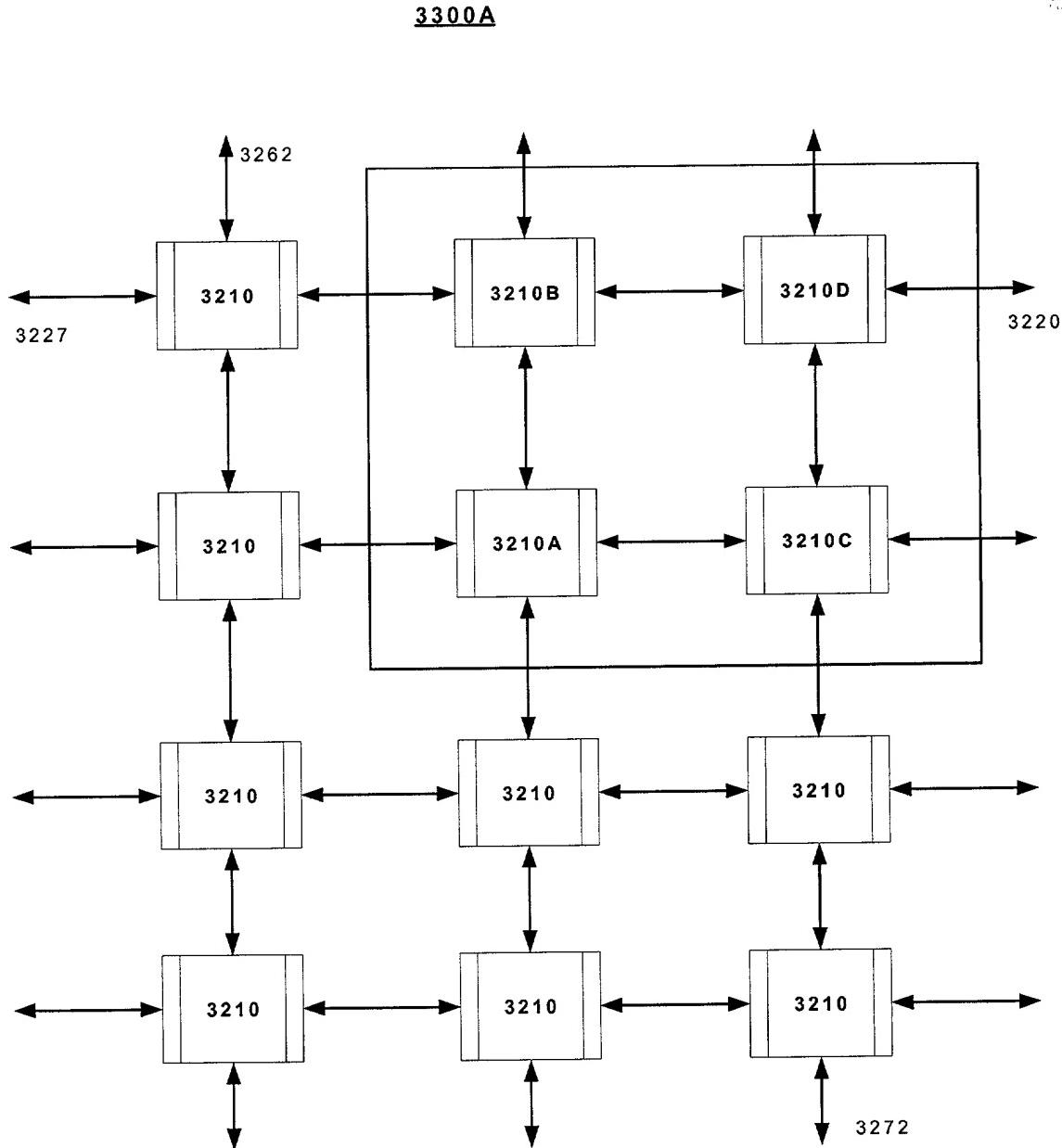
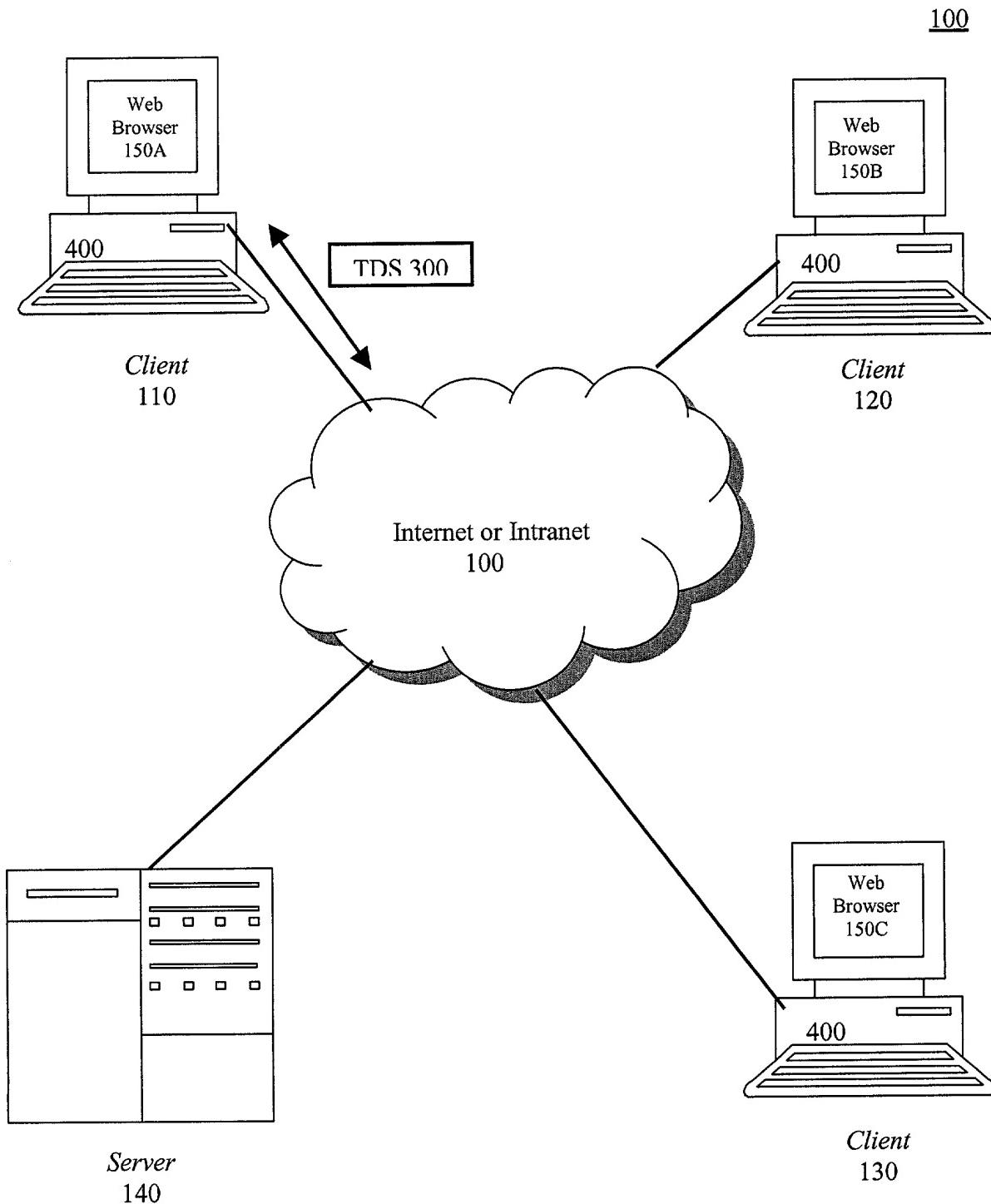
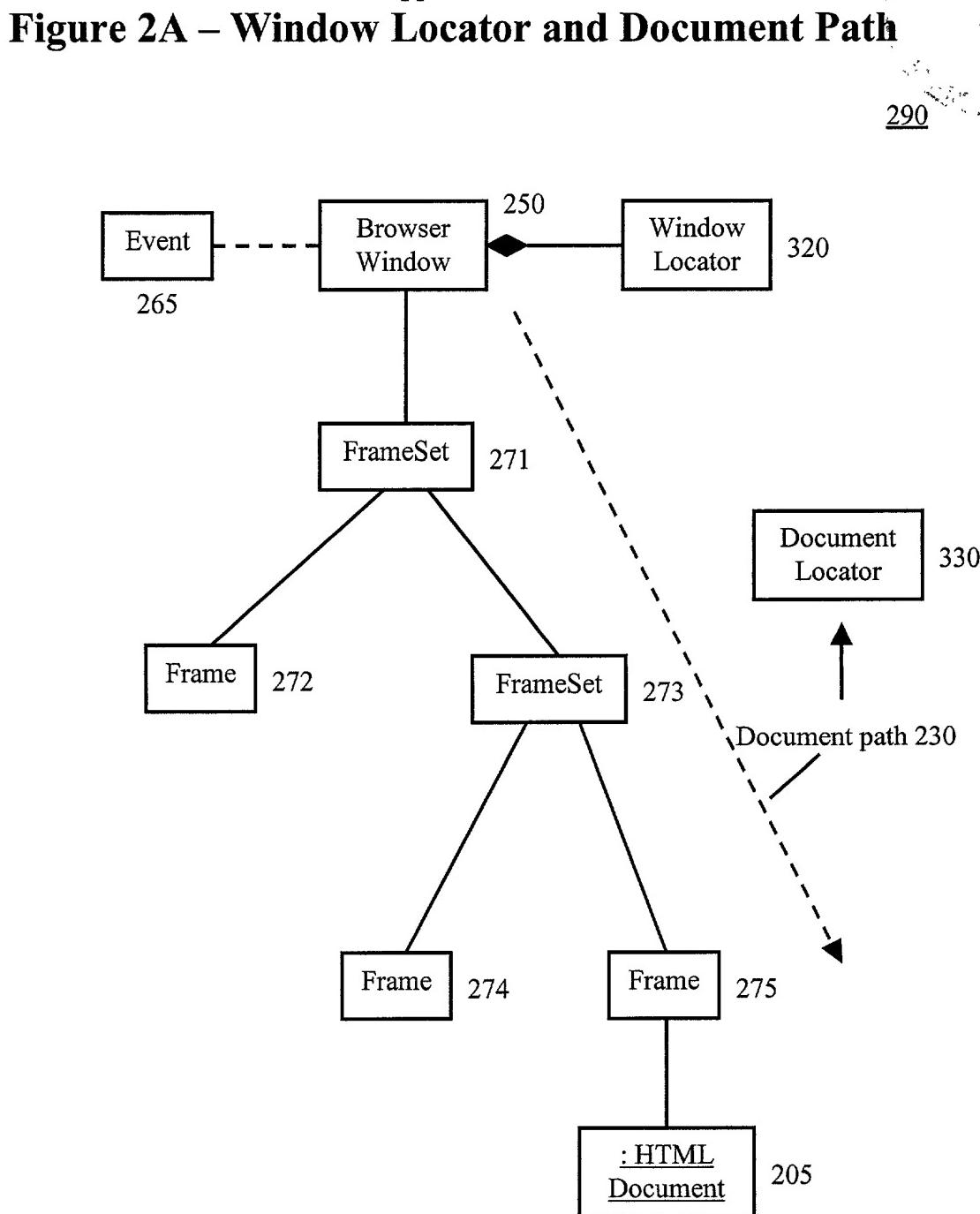


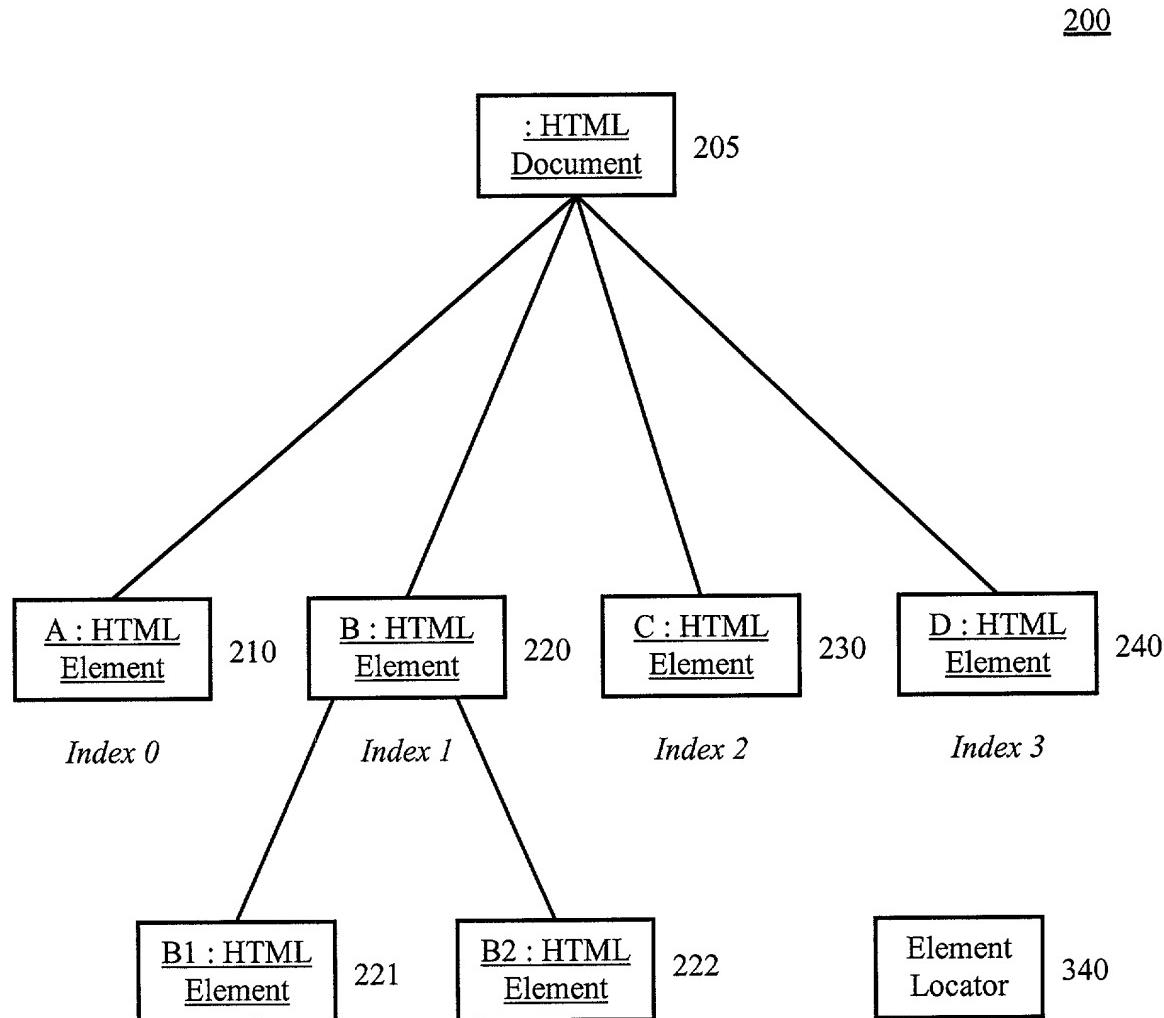
FIGURE 33A

**Figure 1 - Collaboration System Configuration**





## Figure 2B – DOM/DHTML Structure in a Web Browser



**Figure 2C – Element Locator, showing optional mutant web page support**

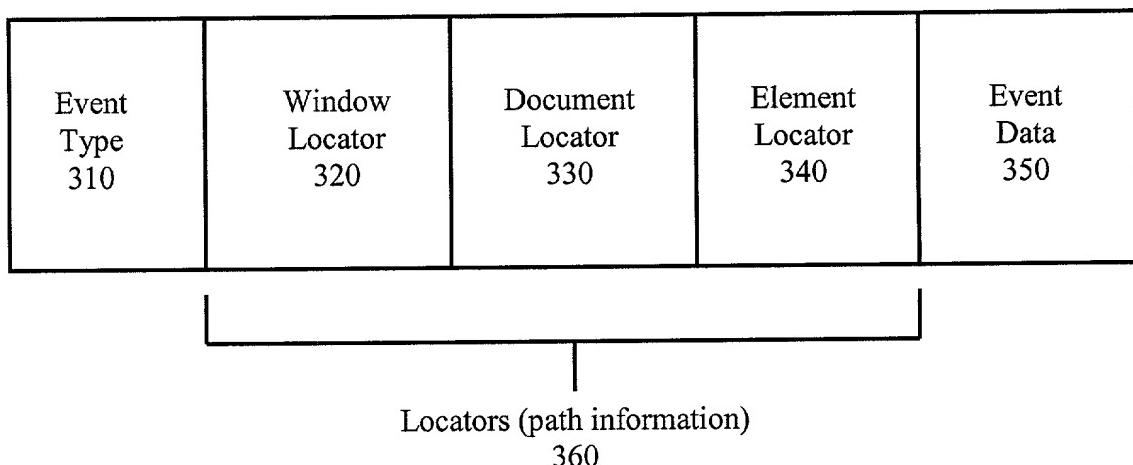
340

Element tagName and index 341	Mutant web page hash value 342
--	--------------------------------------

## Figure 3 – Transfer Data Structure (TDS)

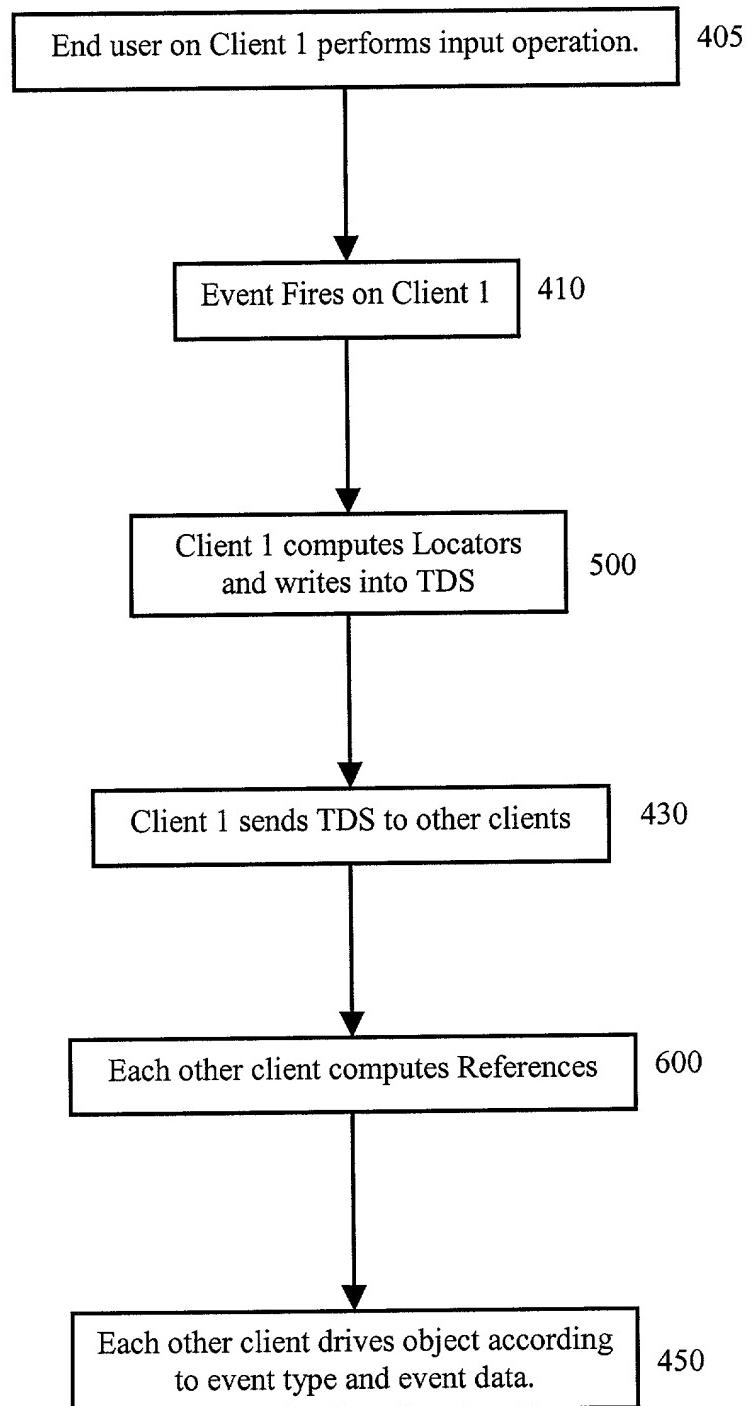
### Transfer Data Structure (TDS)

300



**Figure 4 -- Collaboration System Flow**

400



**Figure 5 – Computing Locators and Writing into TDS**

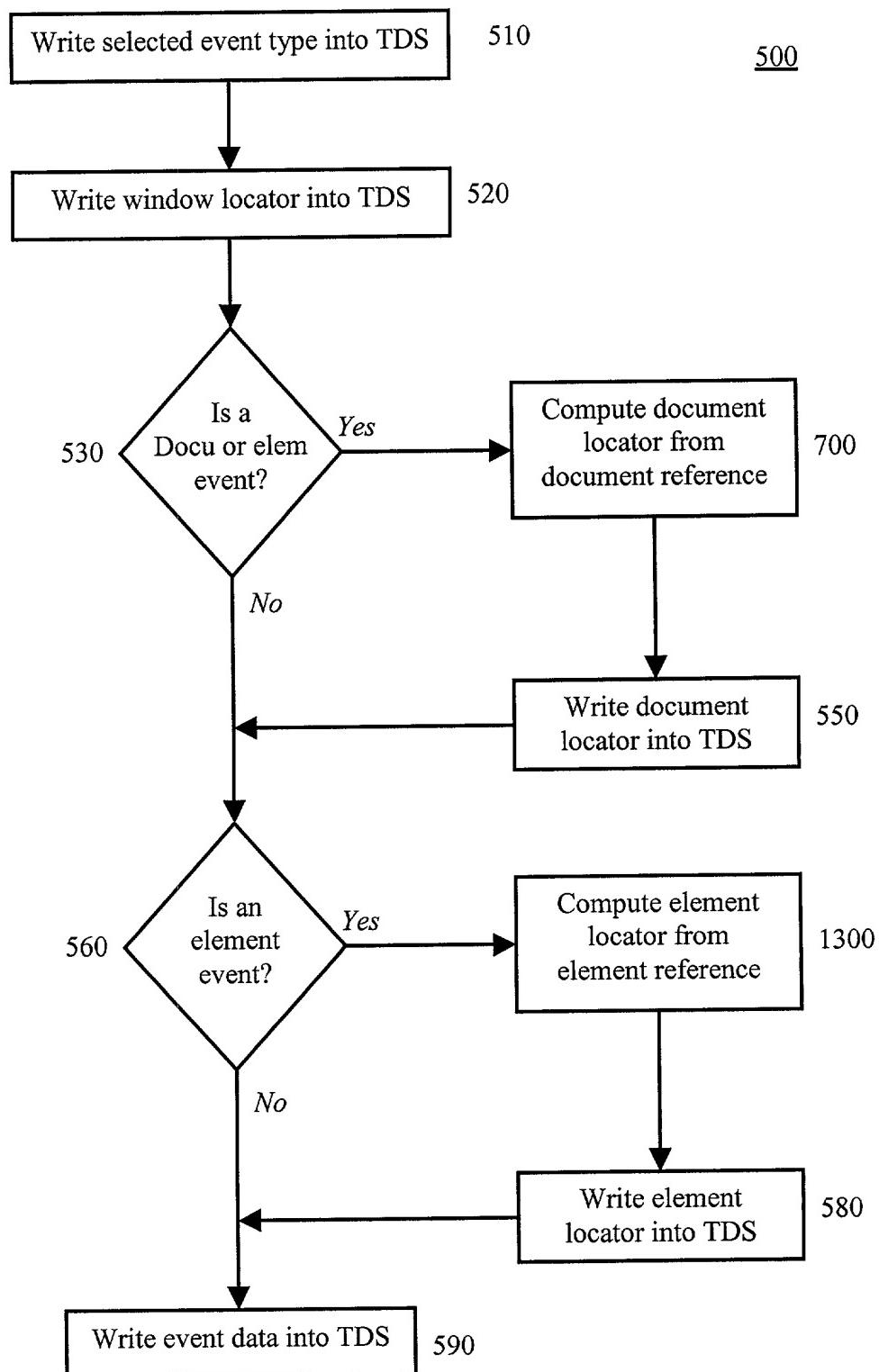
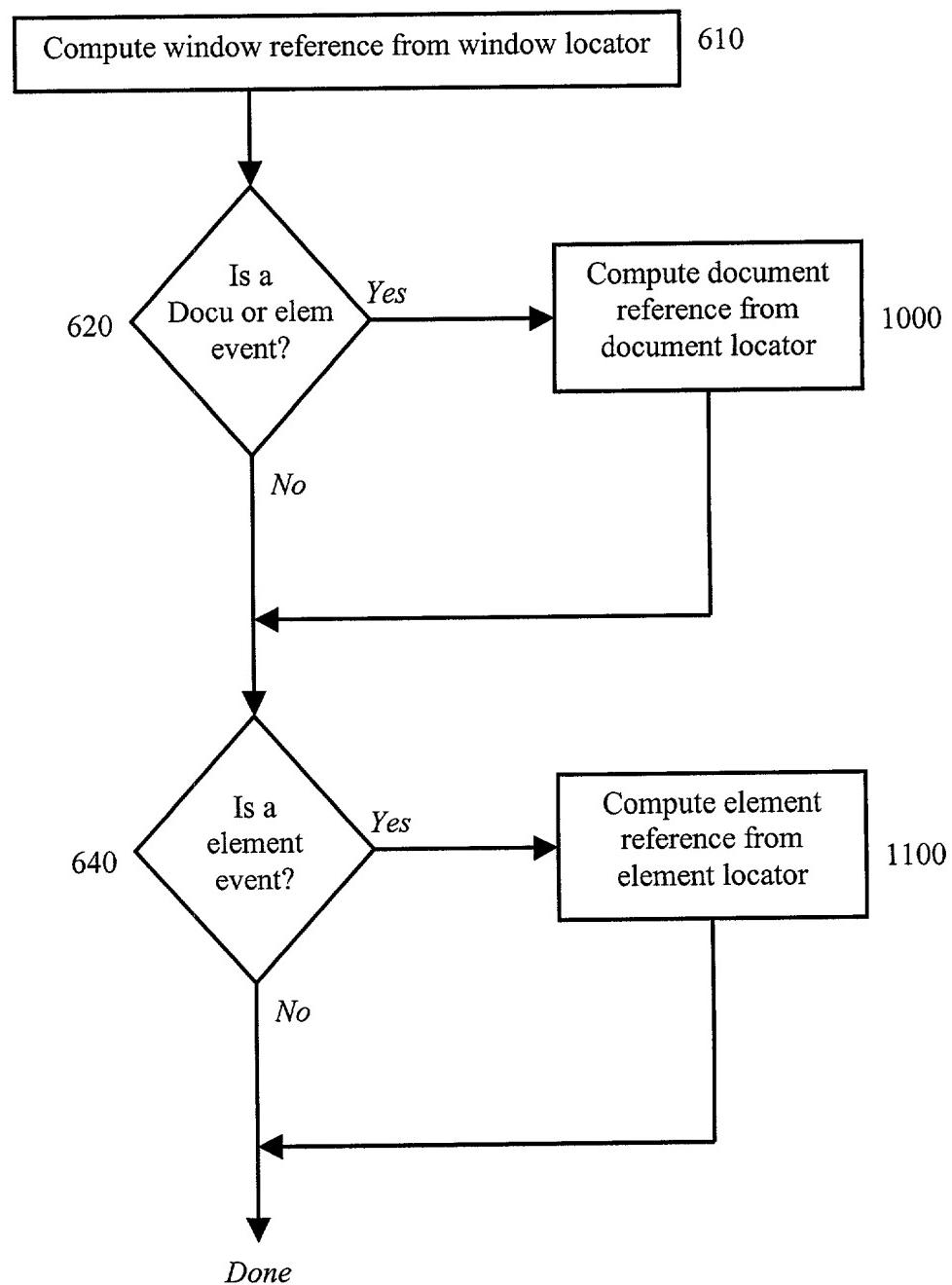
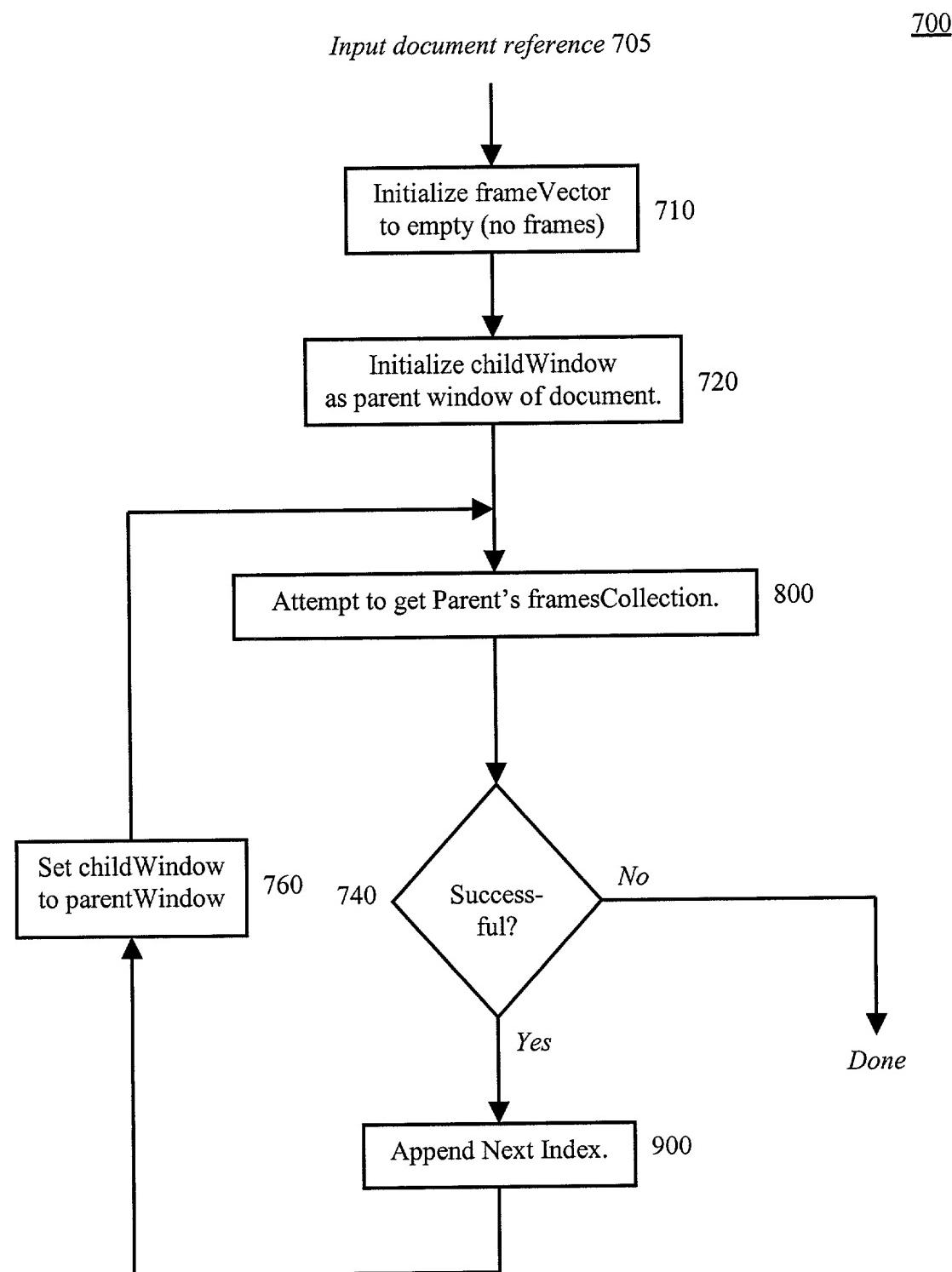


Figure 6 – Computing References

600



**Figure 7 – Compute Document Locator from Document Reference**



**Figure 8 – Attempt to Get Parent's framesCollection**

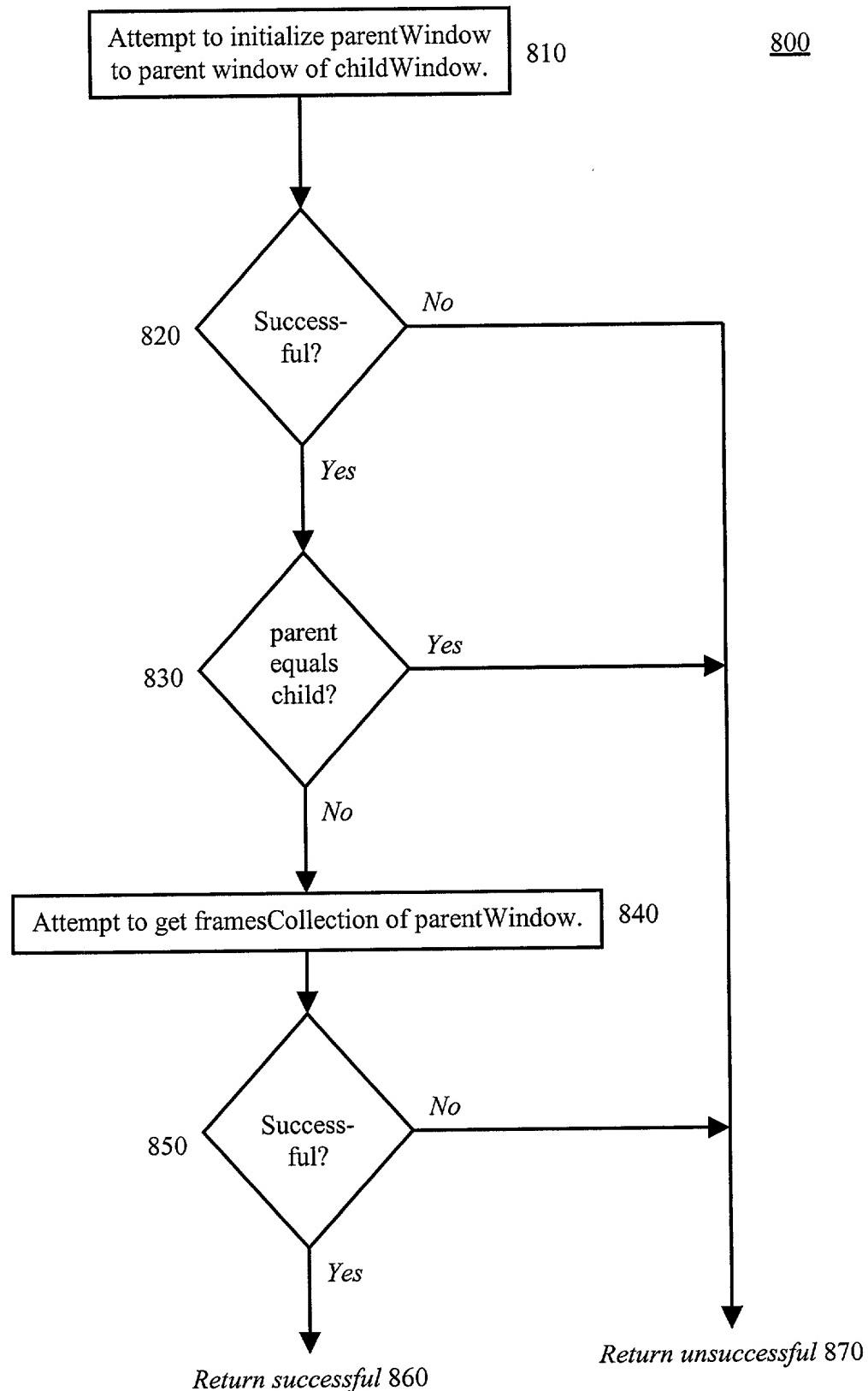
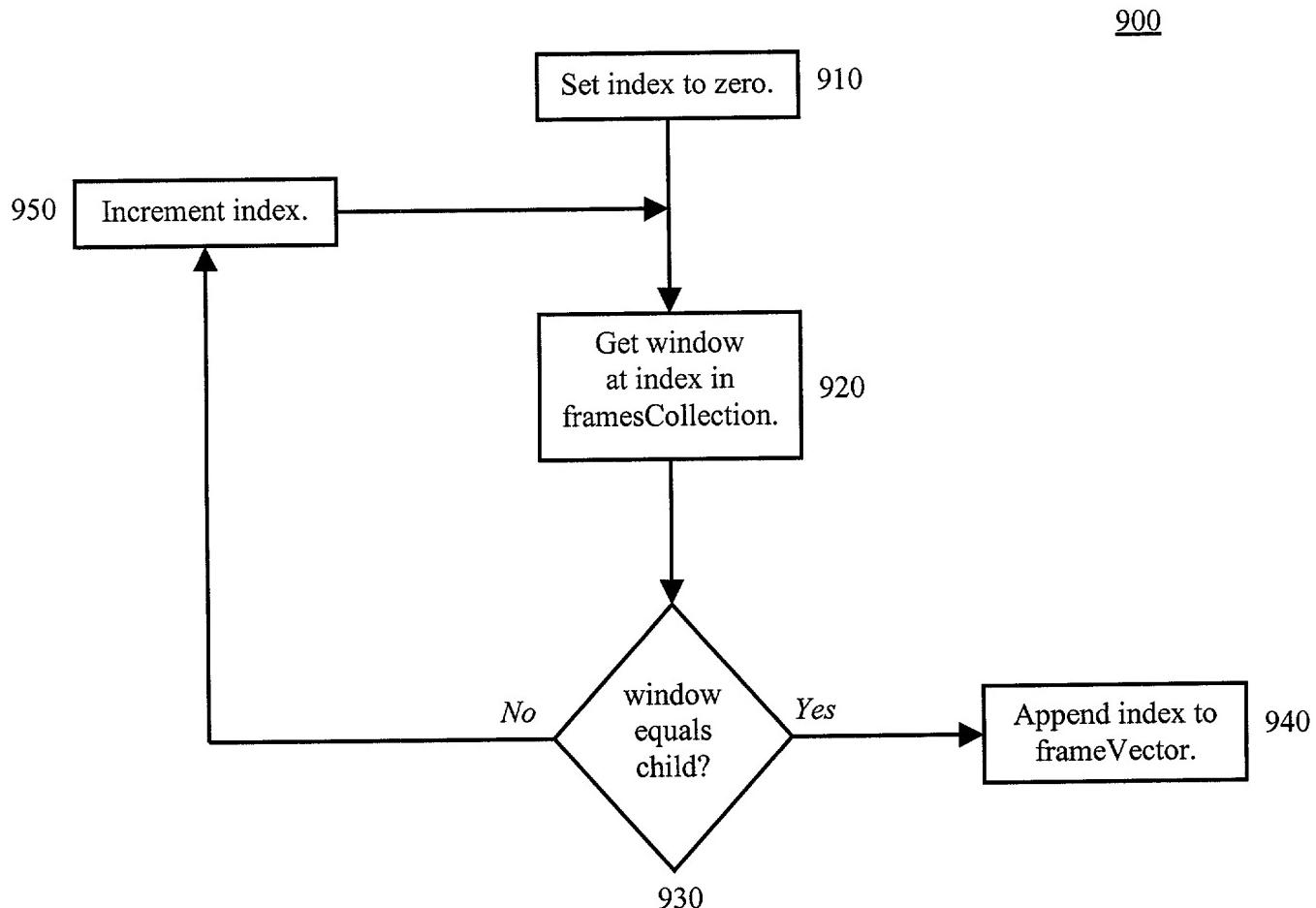
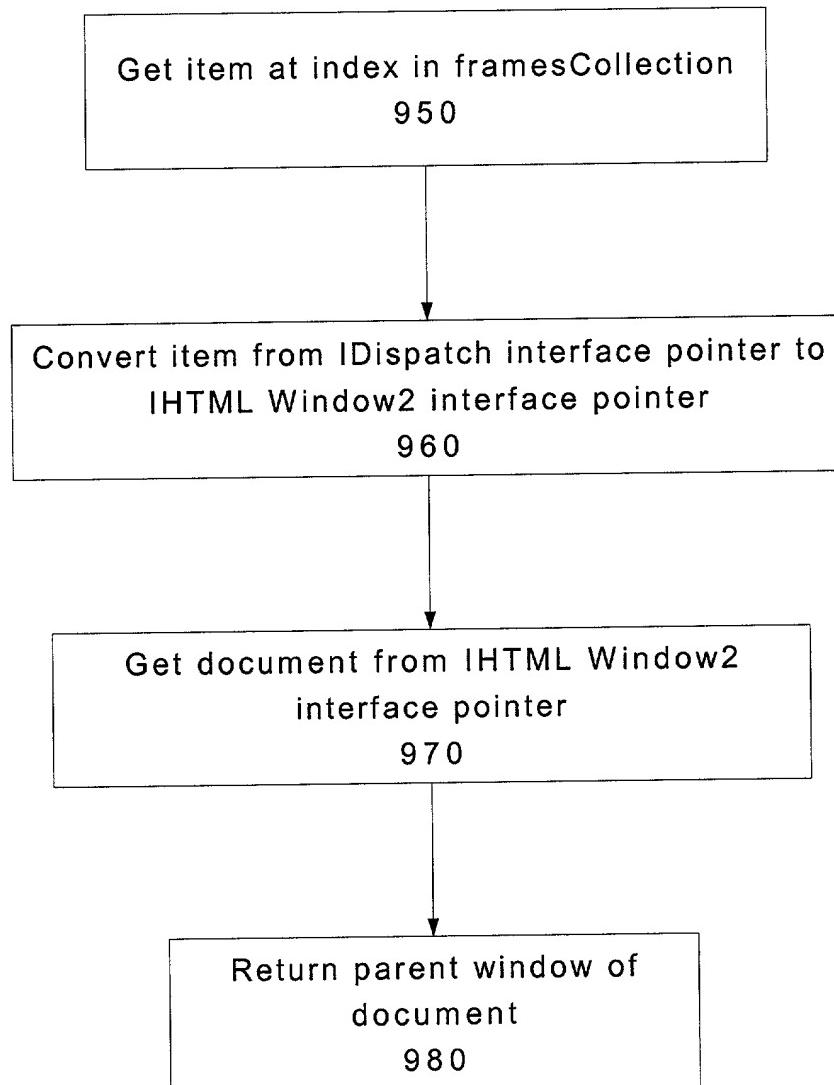


Figure 9A – Append Next Index



One Embodiment Of  
920



Compute Window at Index in IE5  
FramesCollection

Figure 9B

**Figure 10 (Sheet 1 of 2) – Compute Document Reference from Document Locator**

1000

*Input frameVector from Document Locator*

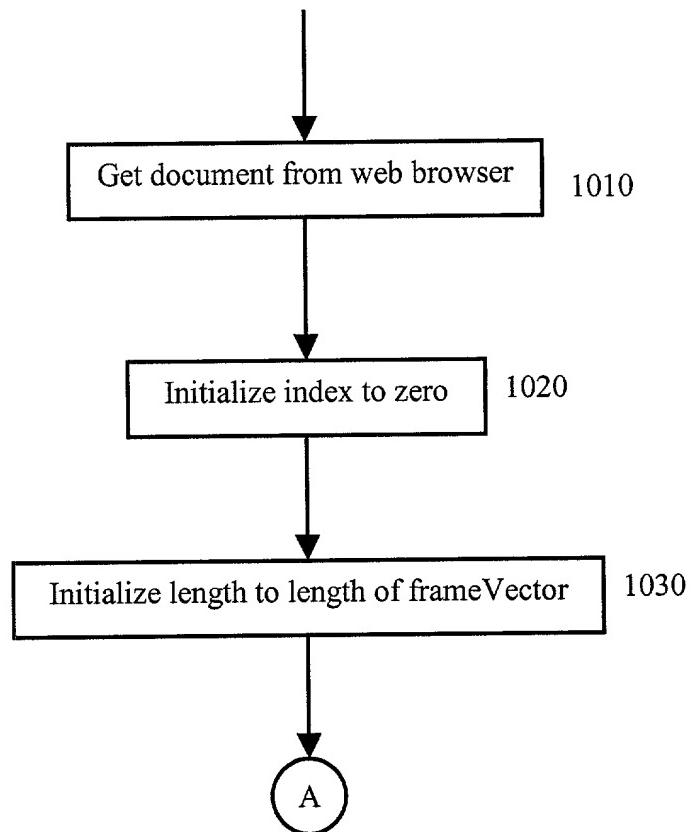


Figure 10 (Sheet 2 of 2)

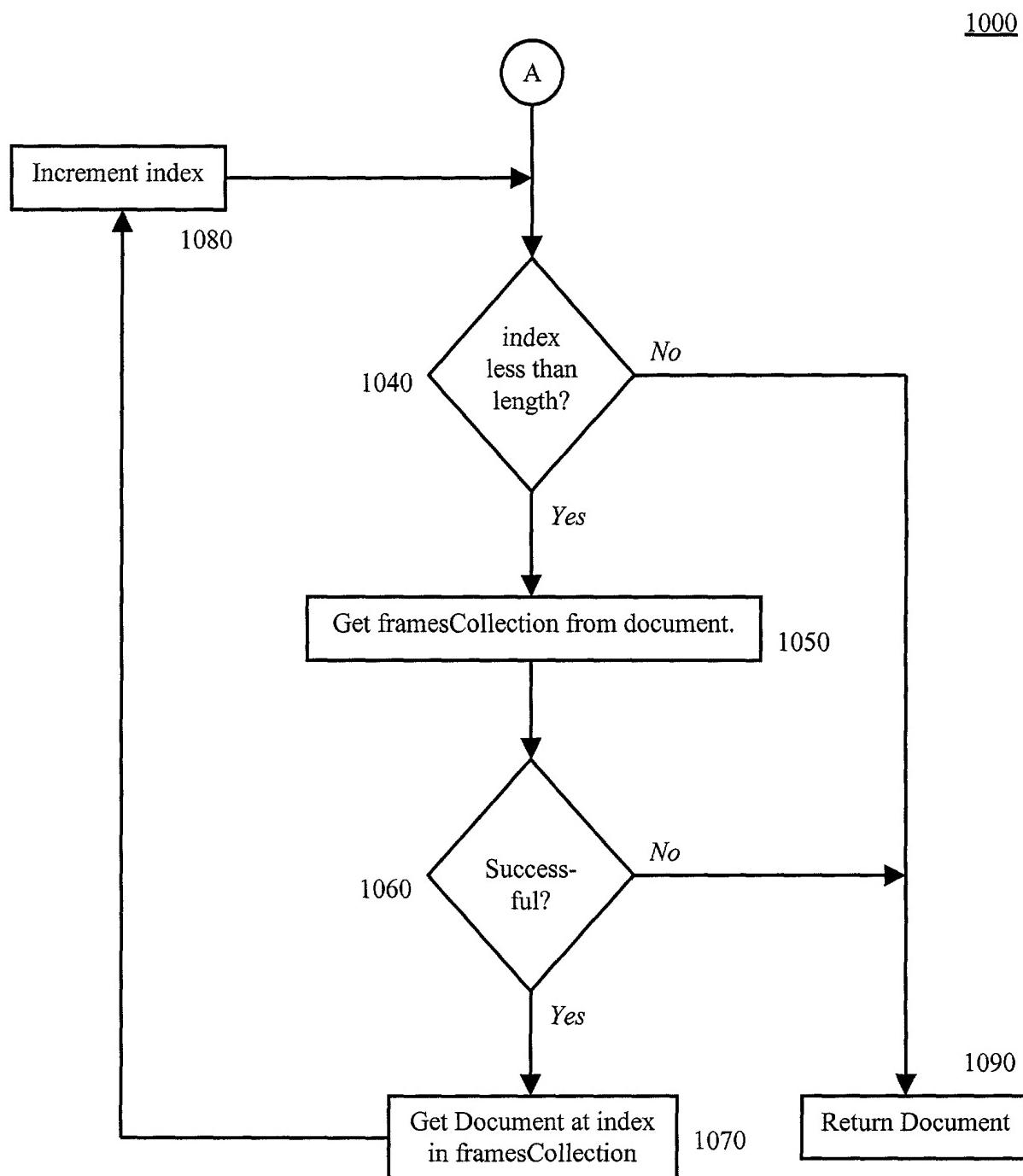
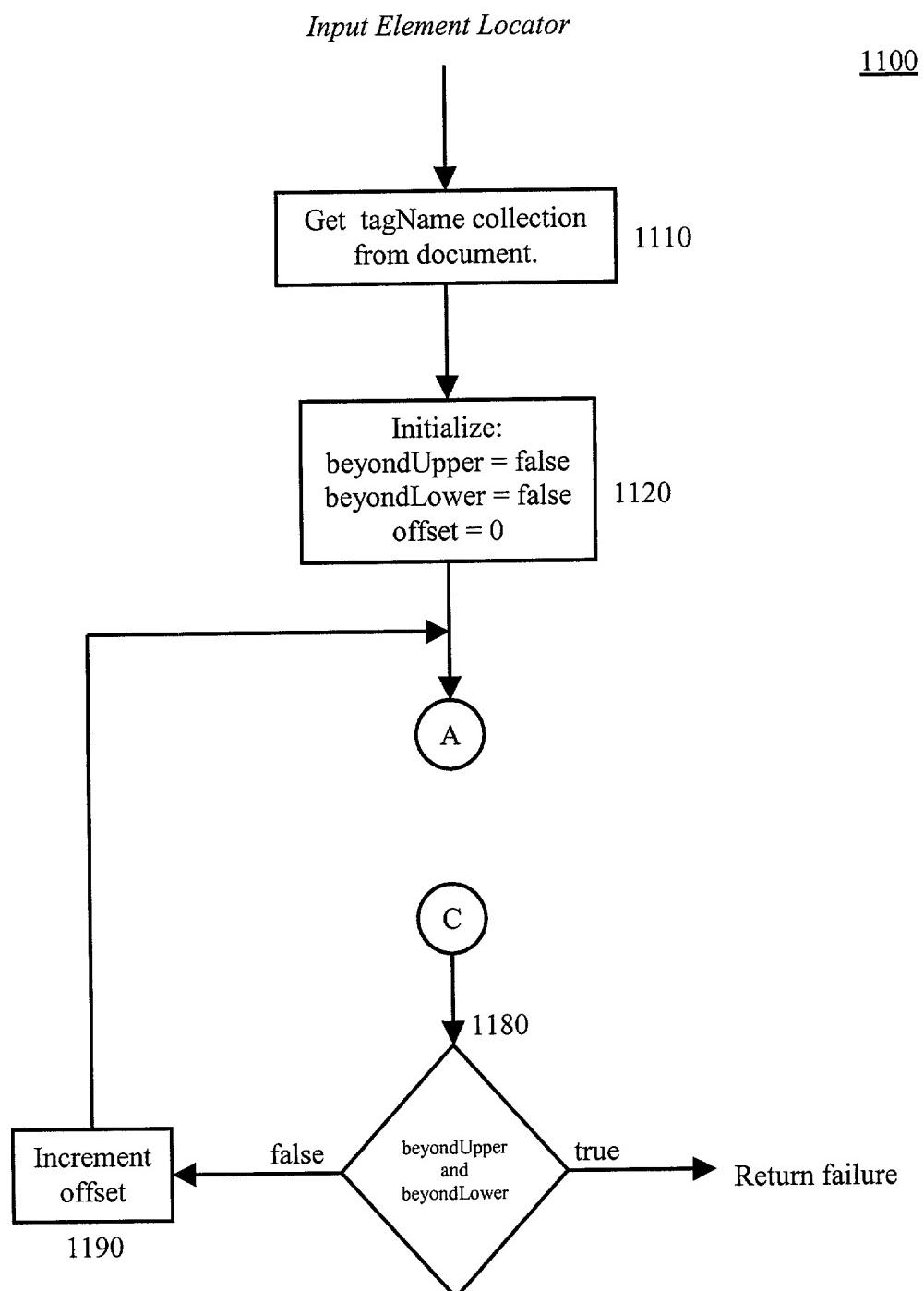


Figure 11 (Sheet 1 of 3) – Compute Element Reference from Element Locator



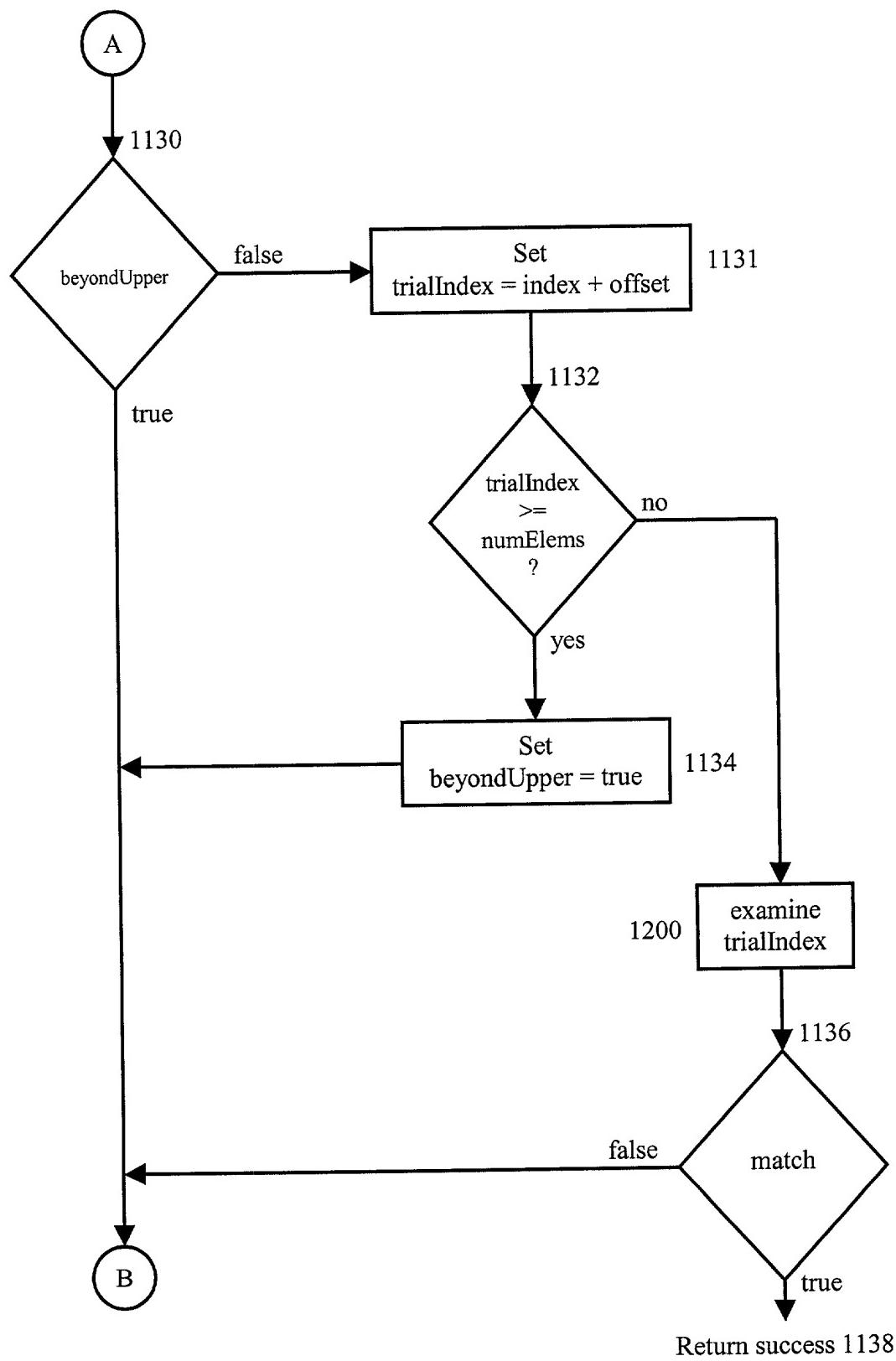


Figure 11 (Sheet 2 of 3)

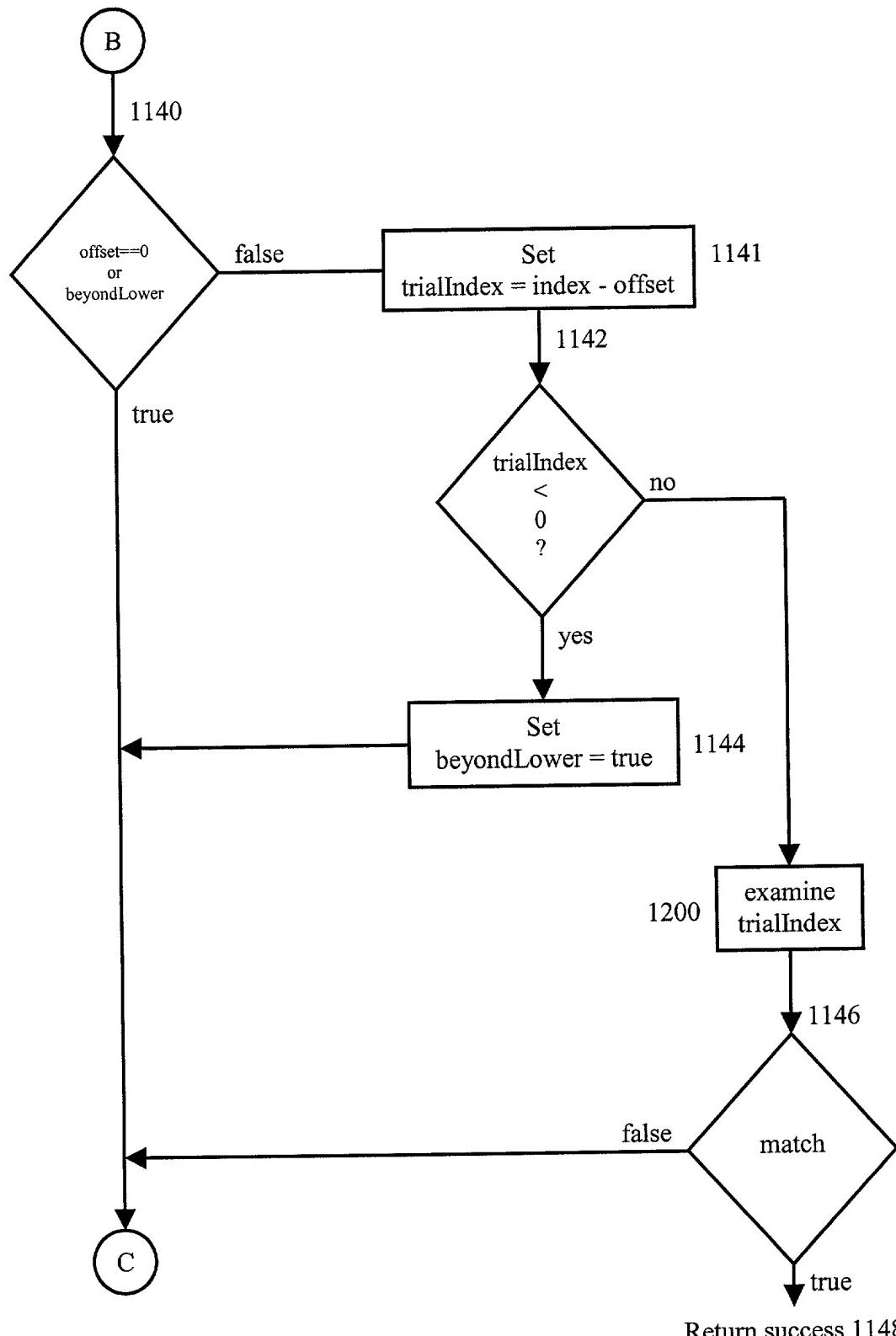
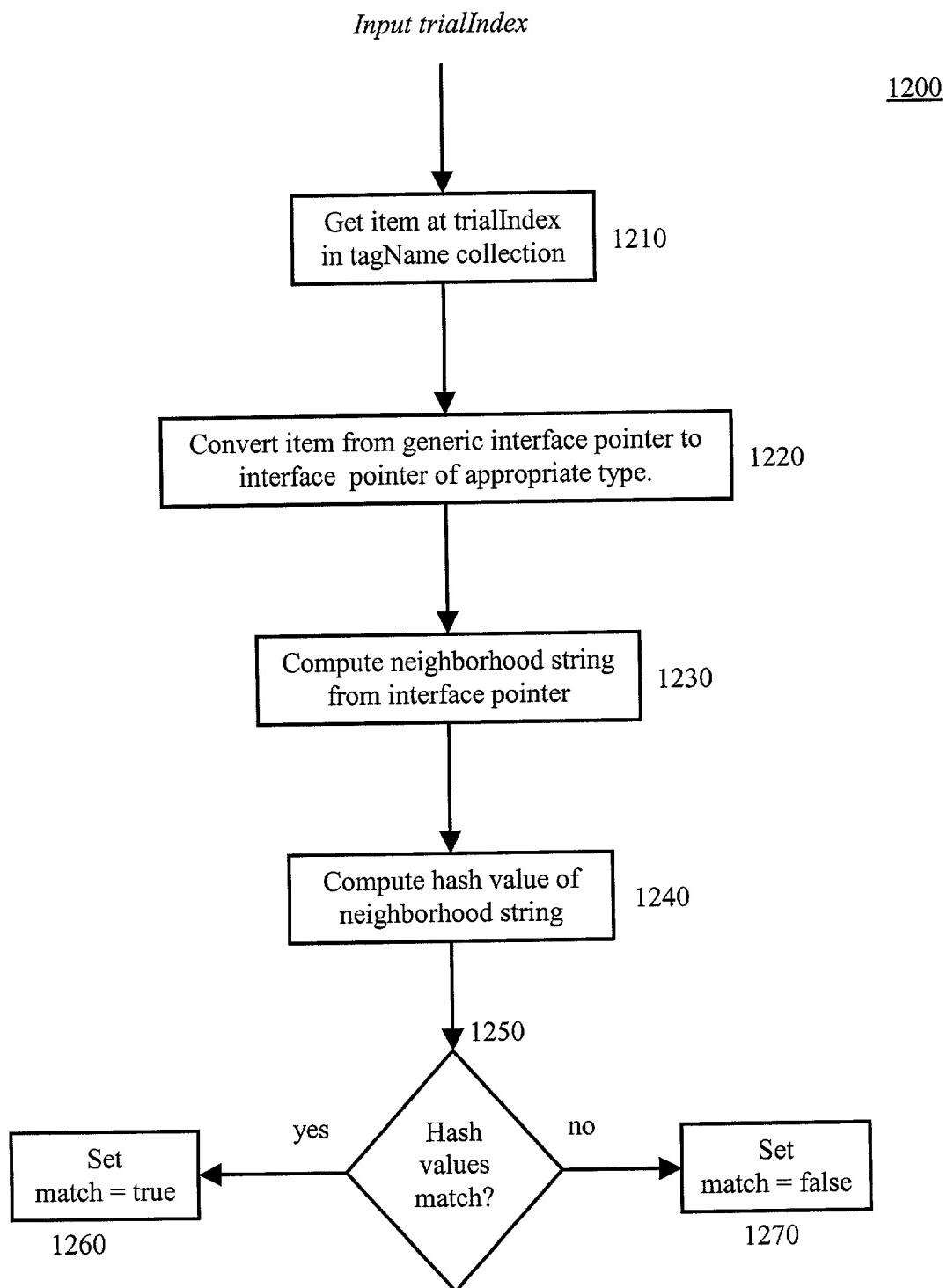


Figure 11 (Sheet 3 of 3)

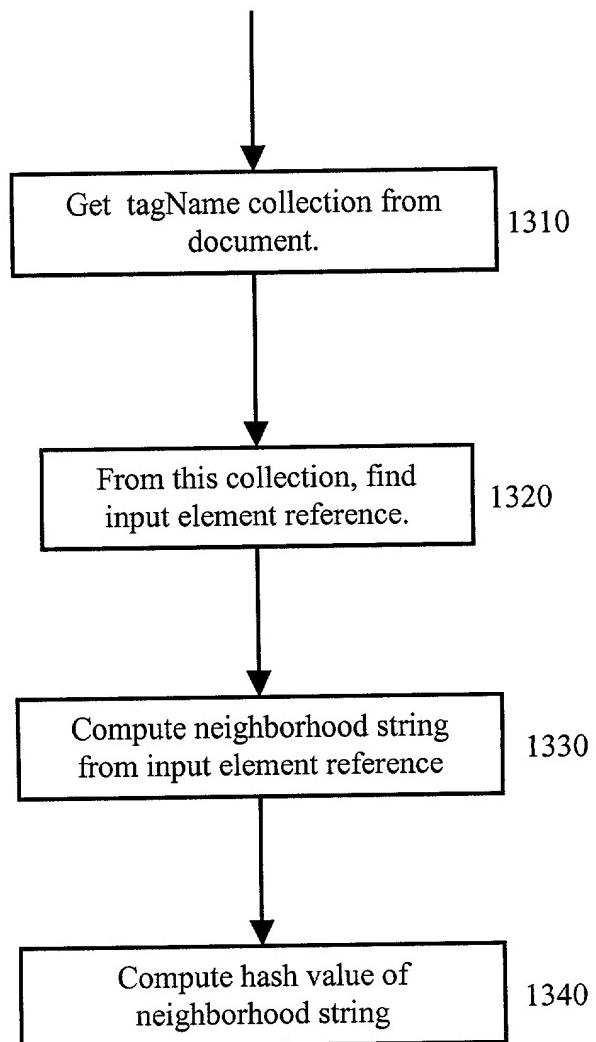
**Figure 12 – Examine trialIndex**



## Figure 13 – Compute Element Locator from Element Reference

1300

*Input Element Reference*



**Figure 14 – Virtual Document Stack**

